

REPORT DOCUMENTATION PAGE

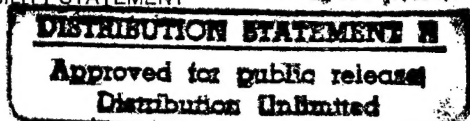
Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 1/26/98	3. REPORT TYPE AND DATES COVERED Final Report 4/25/97-1/25/98
4. TITLE AND SUBTITLE A Case-Based Reasoning Approach to Internet Intelligent Tutoring System (ITS) Authoring		5. FUNDING NUMBERS F41624-97-C-5017
6. AUTHORS Richard H. Stottler		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Stottler Henke Associates, Inc. 2016 Belle Monti Avenue Belmont, CA 94002		8. PERFORMING ORGANIZATION REPORT NUMBER No. 149 Internet ITS Final Report
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Materiel Command 7909 Lindbergh Drive Brooks AFB, TX 78235-5353		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES NONE		

12a. DISTRIBUTION/AVAILABILITY STATEMENT

N/A



19980218 054

13. ABSTRACT (Maximum 200 words)

Report developed under SBIR contract. Intelligent tutoring systems (ITSs) have shown great promise in a variety of training domains and can achieve many of the same benefits as one-on-one instruction, in a cost-effective manner. Our Phase I ITS authoring tool prototype demonstrated an innovative approach to Internet ITS development using Case-Based Reasoning (CBR) paradigm which largely circumvents the difficult and time-consuming process of knowledge elicitation. The case-based approach to knowledge gathering was shown to be highly intuitive, and authoring of the training course was greatly simplified, requiring no special computer training. Phase I also demonstrated critical Internet communication capabilities to allow the ITSs to be delivered, instruct, cooperate, and send back updated student models, over the Internet. The authoring tool prototype was used to develop and demonstrate a prototype AWACS ITS.

In Phase II, we will develop a fully functional Internet ITS Authoring tool, for use by Air Force personnel to develop Internet ITSs which use a CBR approach to train air crew and others. This tool will be applied to the development of an ITS for training AWACS personnel. The Air Force will benefit from improved training capabilities, reduced manpower needs and travel costs, increased flexibility, and better and more efficient training.

14. SUBJECT TERMS SBIR Report, Case-Based Instruction, Intelligent Tutoring System, Training, Course Authoring, Learning, Case-Based Reasoning, Artificial Intelligence, Knowledge Elicitation			15. NUMBER OF PAGES 64
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT

NSN 7540-01-280-5500

Computer Generated

STANDARD FORM 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

DTIC QUALITY INSPECTED 3

AF97-035

A Case-Based Reasoning Approach to Internet Intelligent Tutoring System (ITS) Authoring

Final Report

January 26, 1997

Richard H. Stottler
Principal Investigator

Stottler Henke Associates, Inc.
1660 S. Amphlett Boulevard, Suite 350
San Mateo, CA 94402
(650) 655-7242

DTIC QUALITY INSPECTED 3

Table of Contents

0.0 Abstract

1.0 Summary Report

2.0 Background

2.1 Problem Background

2.2 Artificial Intelligence Methodologies

2.3 Related Work

3.0 Phase I Objectives/accomplishments/Results

4.0 Phase II IITSAT

4.1 System Functionality

4.1.1 Intelligent Tutoring Systems Overview

4.1.2 ITS Authoring Tool Overview

4.1.3 Internet Delivery of IITSAT and ITSs

4.2 Design

4.2.1 General ITS Architecture

4.2.2 Internet ITS Authoring Tool (IITSAT)

4.2.3 Internet ITS Server

4.2.4 AWACS ITS

5.0 Phase I Prototype

5.1 Functionality

5.2 Demo Sequence Overview

6.0 Future Work

6.1 Phase II

6.2 Phase III

References

Appendix: Demonstration Screen Dumps

0.0 Abstract

Intelligent tutoring systems (ITSs) have shown great promise in a variety of training domains and can achieve many of the same benefits as one-on-one instruction, in a cost-effective manner. Our Phase I ITS authoring tool prototype demonstrated an innovative approach to Internet ITS development using a Case-Based Reasoning (CBR) paradigm which largely circumvents the difficult and time-consuming process of knowledge elicitation. The case-based approach to knowledge gathering was shown to be highly intuitive, and authoring of the training course was greatly simplified, requiring no special computer training. Phase I also demonstrated critical Internet communication capabilities to allow the ITSs to be delivered, instruct, cooperate, and send back updated student models, over the Internet. The authoring tool prototype was used to develop and demonstrate a prototype AWACS ITS.

In Phase II, we will develop a fully functional Internet ITS Authoring tool, for use by Air Force personnel to develop Internet ITSs which use a CBR approach to train air crew and others. This tool will be applied to the development of an ITS for training AWACS personnel. The Air Force will benefit from improved training capabilities, reduced manpower needs and travel costs, increased flexibility, and better and more efficient training.

1.0 Summary Report

The Air Force training budget is significantly burdened by travel costs. Typically, training involves transportation of the trainees to the training site and lodging while there. The Air Force needs a cost-effective way to deliver training to where the trainees are located. Further, in complex domains, instruction is often complicated by the need for the student to master a variety of concepts and to apply them in unique situations and in different sequences. In these kinds of domains, the student must develop a competence not only in the relevant facts and skills, but also an understanding of the concepts underlying these procedures.

Recent advances in Internet technology provide the unique opportunity to distribute Air Force training across multiple sites and dramatically reduce travel-related training costs. Not only can students receive training at their own local sites, but instructors can monitor students' progress from a distance, and course authors can maintain and update training material across the Internet. Additionally, an ITS can promote instructor productivity, cope with the higher skill requirements for trainees, provide tailored instruction and remediation, while allowing flexibility in teaching methods and achieve many of the same benefits as one-on-one instruction.

During Phase I, we completed all 6 Phase I technical objectives described in the Phase I proposal. The first result was the concept of the "loose confederation of ITSs" idea. The inspiration for this original idea came from the world wide web and its' current usage. People browse one web page, which may have links to other web pages, which they may follow and visit, even though the authors of the separate pages are probably unknown to each other. Further, search engines exist to help guide end-users to pages which may be of interest. Our concept for ITSs follows naturally from this model.

In Phase I, we also designed and implemented a scheme which would allow the required communication between ITSs. In our concept, a central ITS server keeps a master list of all ITSs, their capabilities, and a global list of principles. The ITS authoring tool running locally on the author's machine must be able to get information from and transfer information back to the central ITS server enabling the author to send course updates and the instructor to get student models. Finally, the cooperative ITSs must be able to send student models to each, launch each other, and get information about each other.

We developed and implemented a very general structure to represent cases, applicable across several domains. This structure represents examples, exercises, and scenarios for instructional purposes in a common format. Each includes: (1) a multi-media description of the problem, which may evolve over time (as in tactical scenarios); (2) a description of the correct actions to take possibly including order-independent, optional, and alternative steps; (3) multi-media explanations of why these steps are correct; (4) the list of methods which determine whether the steps have been correctly executed by the student; and (5) the list of principles required to know the correct action to take.

We developed a general ITS Architecture and implemented a simplified version of it in Phase I. This architecture addresses aspects of instruction such as assessment, remediation, use of examples, and style of course presentation. The Phase I prototype monitors student actions in scenarios, derives a model of the student's knowledge, selects the most appropriate examples for instruction and exercises for testing, and selects the Topics that address the areas that the student doesn't understand. It also provides a facility for the student to browse those topics himself. Based on the knowledge requirements of ITSs represented in this architecture, we designed and implemented a simple version of the ITS authoring tool.

We held discussions with several members of the AWACS community to determine their most pressing training requirements. The number one concern of all AWACS WDs is Radio Transmission (RT), that is, communication with other aircraft, especially fighter pilots. To address this most pressing concern in Phase I, we implemented the AWACS WD communication ITS, using the prototype authoring tool. Based on our experience with developing the AWACS ITS prototype, we refined the authoring tool prototype. In December, we demonstrated the authoring tool and AWACS ITS to Armstrong Laboratory.

The ITS interacts with the student, monitoring his actions and the information that he is reviewing. Based on this monitoring, the ITS maintains a model of the student's knowledge. It uses this model, along with pedagogical knowledge and student input, to decide what the next type of instruction should be. It might be to display an example relevant to the principles being taught at the time. It may be to test the student with an exercise, based on his student model, using principles that the ITS believes he may have learned. It may be to debrief the student on the mistakes of his last exercise. Or it may be to formulate a remedial course of instruction, based on the deficiencies in the student's current model. These remediations may take the form of examples or general topic information.

The ITS Authoring Tool is designed to allow a course instructor to easily enter domain and other knowledge without requiring computer programming skills. The Internet ITS Authoring Tool (IITSAT) then generates an ITS for the particular course. The IITSAT must facilitate the entry of examples/exercises, including problem description, solution, solution steps, and explanations. The examples may be in the form of scenarios or simulations. It must allow entry of the course principles in an organized fashion. It must be able to integrate Internet multi-media courseware (developed with well-known authoring tools) especially when that courseware includes descriptions of the principles or provides motivational passages. In addition to course knowledge, the instructor can enter other domain knowledge, pedagogical knowledge (how best to teach), and student modeling knowledge (how to assess actions and how to determine mastery). Defaults would typically be provided for these types of inputs.

Our Internet ITS Authoring Tool (IITSAT) will allow easy development and maintenance of training systems which are based on pedagogically sound instructional strategies, validated through empirical research. The case-based approach, using scenarios and simulations to teach as well as assess student knowledge, is such an approach. The training systems created with the tool will include a variety of media, including graphics, Java scripts, HTML, video, sound, animation, and simulation.

In general, the Phase I prototype implements all aspects of the full-scale system, though in a simple form and not necessarily integrated with each other. That is, the prototype includes representative ITS, ITS Authoring, and Internet functionality. Additionally, the prototype authoring tool was used to create a prototype AWACS ITS.

ITS

The Phase I ITS prototype presents a scenario to the student, monitors his actions in the scenario, compares those to an expert's description of the correct actions, then determines what principles the student applied correctly and incorrectly. It then debriefs the student, updates the student model with those lists and shows the student appropriate remediation. It then selects another scenario based on what principles the student needs testing on.

ITS Authoring

The authoring tool prototype allows users to enter the knowledge needed by the ITS. The prototype allows entry of the Principle hierarchy (an example is given on the next page), Topic structure, and scenarios. Topics may consist of files containing multi-media.

Internet

Using CGI scripting and JAVA programming, we implemented a way for an ITS running as a Web page on a Web server to pass information to another ITS running as a Web page on another Web server. We have also implemented a way for an ITS, running either locally on a PC with a current Internet connection or as a Web page on a Web server, to request and pass information back and forth to a central FTP site.

2.0 Background

2.1 Problem Background

The Air Force training budget is significantly burdened by travel costs. Typically, training involves transportation of the trainees to the training site and lodging while there. The Air Force needs a cost-effective way to deliver training to where the trainees are located. Further, in complex domains, instruction is often complicated by the need for the student to master a variety of concepts and to apply them in unique situations and in different sequences. In these kinds of domains, the student must develop a competence not only in the relevant facts and skills, but also an understanding of the concepts underlying these procedures. Instructional courses must be attuned to the trainee's background and needs, motivate him to develop an accurate and thorough understanding of the subject matter, and then effectively verify the correctness of his understanding and remediate inaccuracies.

When students are required to be flexible in their understanding of principles and potential applications the most effective teaching strategy is one-on-one interaction to maximize the role of the teacher. The two-sigma problem (Bloom 1984) describes the fact that students receiving one-on-one instruction perform two standard deviations better than students receiving conventional instruction; the problem is to realize these benefits without requiring an equal number of teachers and students. One-on-one instruction maximizes the adaptability of the instruction process to help the student construct and test a mental model on different circumstances. The student can ask specific questions of the instructor, and the instructor can respond with immediate answers and supporting examples tailored to the individual student. This interaction is also effective because the teacher can gauge the student's learning speed and performance and tailor the introduction of future concepts accordingly. The instructor can also ask specific questions and present specific examples or tests to counter suspected inaccuracies in the student's mental model.

Unfortunately the financial and human resources are simply not available to provide this kind of one-on-one instruction for many complex domains. Weapons systems and computer systems are becoming increasingly complex at a time when downsizing is forcing the Air Force to reduce its training budget and staff. The Air Force must do more with fewer resources.

2.2 Artificial Intelligence Methodologies

Case-Based Reasoning

Case-Based Reasoning (CBR) is the field of AI which deals with the method of solving a current problem by retrieving the solution to a previous similar problem and altering that solution to meet the current needs. Case-Based Reasoning is a knowledge representation and control methodology based upon previous experiences and patterns of previous experiences. These previous experiences, or "cases" of domain-specific

knowledge and action, are used in comparison with new situations or problems. These past methods of solution provide expertise for use in new situations or problems. From our previous ITS experience, we believe that the general problem of teaching students is well suited for the application of a Case-Based Reasoning method.

CBR systems offer enormous benefits compared to standard AI approaches. The knowledge elicitation bottleneck is largely circumvented. Cases can be automatically acquired directly from domain experts. Rules, on the other hand, almost always require the intervention of a knowledge engineer. Instead of having to elicit all of the knowledge required to derive a solution from scratch, only the knowledge required to represent a solution is needed. So knowledge elicitation is largely avoided with CBR and may be COMPLETELY automated depending on the type of application and the expert. This makes CBR especially appealing for an instructional design framework that will potentially be applied to multiple domains because it reduces the knowledge engineering time requirement.

Conventional knowledge base technology dictates a single, fixed problem solving methodology. With CBR, each case (in the extreme), can represent a different methodology. This is important for complex domains where different problems or situations, although sharing the same fundamental concepts, may require different solution strategies.

Intelligent tutoring is a natural application for CBR. Students learn best from examples. These examples consist of a problem, its solution, the problem-solving process, and an explanation as to how that solution was derived. This proposal seeks to aid that natural process through the use of a general CBR system that can document examples (problems, their solutions, the process, and explanations) and the principles and procedures needed to solve them. Teaching will be accomplished through the presentation of appropriate examples and the corresponding course materials. Testing is performed by presenting the problem part of an example and comparing the student's solution to the stored solution.

Stottler Henke Associates, Inc. (SHAI) has performed six projects which developed training systems based on CBR, most including simulations, both existing and new. One project dealt with a team-oriented domain. Our extensive experience with ITSs and group problem-solving can be applied to this project.

Knowledge Elicitation / Curriculum Development

In developing an intelligent tutoring system, the authoring tool would first query the domain experts for exercises (examples). Each example elicited will consist of three main parts: the problem, the solution, and the process of deriving the solution, along with explanations of each step of that process. The problem part is an explanation of the problem to be solved and may be partly graphical in nature, especially for complex subjects. The solution may be a simple one or two word answer, or complex such as a paragraph, diagram, or the correct problem-solving actions to take in a complex physics exercise. The solution process will likely be the most complicated part of all. It consists

of the steps required to solve the problem. With each step is a reference to the general principles or methods used in that step. Each reference points to a principle or method in the body of knowledge that the student must learn. Any principle could be referenced many times in different examples, but that principle would only be represented once in the body of knowledge. A detailed explanation of the referenced principles and problem solving methods could be requested from instructors, thus automatically extending the ITS. The examples force the instructors to include only and all information required for problem solving. A reasonable organization is mapped onto (possibly) unorganized experts.

Interactive Multi-Media

Interactive multi-media is used to present both the course material and the examples. SHAI has experience in the following media:

- | | |
|--------------------------------------|--------------------------|
| - Interactive Simulations | - Interactive Animations |
| - Interactive 3 Dimensional graphics | - Interactive Photos |
| - Audio | - Virtual Reality |
| - Video | - Hypertext |

Interactive simulations are especially useful as examples for a course. For example, in the Astronaut domain, we used a simulation of an experiment to show the effect of the astronaut's actions on the experimental results. The student could vary his action within the example and see the effects. This effectively allows one example to behave as many and allows students to ask what-if questions. Rerunning the simulation allows an immediate answer to the questions. In our previous simulations work, we have developed tools and techniques for the rapid development of object-oriented simulations.

Another technique, related to simulations is interactive animations. In the example above, instead of just printing the results of the simulation, the experimental device is animated. Seeing the results makes the example much more vivid and therefore more likely to be remembered. SHAI has already developed a general three-dimensional animation capability.

Interactive 3 dimensional graphics is another important tool. In many applications, both situational and equipment-oriented, three dimensional visualization is required. The view can be rotated by the students to gain a clearer understanding. Other computer generated graphics such as bar charts, pie charts, bitmap files, line graphs, and plots can be supported. SHAI has already developed an interactive three-dimensional graphics capability.

An extension of the interactive simulations, animations and 3-dimensional graphics is virtual reality technology. Through the use of head mounted displays or goggles; hand, finger, and body tracking; and three-dimensional sound, an operator can achieve a more

realistic, immersive experience in the tactical simulation. SHAI is currently involved in two virtual reality projects.

2.3 Related Work

Within the last few years, CBR has emerged from its academic setting ([Schank], [Kolodner]) and gained wider recognition for its usefulness and applicability to many problems in practical domains. Recent articles describe the relationship between CBR and human reasoning and the various techniques for organizing case-bases and providing timely and relevant retrieval of stored cases. In addition, there are now three seasoned CBR application development tools on the market: ESTEEM, ReMind and CBR Express. The introduction of these shells indicates the increasing popularity of a CBR approach to problem solving.

The application of CBR techniques to ITS design is only in its infancy. [Edelson] presents CreANIMate, a biology tutor that uses CBR to index relevant stories to principles of animal morphology. [Rosenblum and Moore] describe a CBR approach to the construction of explanations for the ITS called SHERLOCK. [Schult] describes CABAT the case-based tutor which can retrieve prior similar situations to show the student. [Aleven and Ashley] describe the use of CBR to automatically generate contexts (sets of related legal cases) for use by a human tutor. The work performed at SHAI has done much to further the use of CBR in ITS development.

[Towne & Munro 1989] describes a system called RAPIDS (RAPid ITS Development System) which attempts to provide high cognitive fidelity of simulation. It has two goals - to provide effective instruction over a wide range of devices and task types, and to provide the tools for developing new applications in a cost-effective manner. It is similar to our approach in that it emphasizes the use of simulations. It's different in that it still primarily addresses training from a task perspective, where we concentrate more on principles and student modeling. It has been primarily applied to diagnostic domains while our work has tended to concentrate on tactical ones. RAPIDS has been applied to one tactical domain but it has not been applied operationally. The Navy considers it still primarily a research system. The RAPIDS approach depends very strongly on an accurate model of the device. Otherwise students will learn how the simulated item behaves and not the real one. Our approach is not so dependent on simulator fidelity, relying somewhat more on the fidelity of the scenarios.

[Regian and Shute 1994] describes MHO, a tutor that teaches basic principles of electricity. From experimentation it was found that students with more exploratory behaviors learned significantly faster and scored significantly higher on outcome tests than if they had been restricted to one environment. On the other hand, less exploratory learners performed significantly better in a more structured environment, thus indicating the benefits of adapting to student behavior. We have found similar results in operational environments. In the realm of sonar interpretation of acoustic features, when an operator detected a foreign submarine, he could go back and explore the previous 24 hours worth

of tapes to see if there were any indications of the submarines' presence earlier. In this way new features that indicate different types of submarines could be found. OTACS Chief Tim Richey in particular excelled at this type of exploration and became one of a few acknowledged experts. Ninety percent of the technicians, however, do not exhibit this behavior and never reach a high state of expertise.

[Shute et al. 1989] describes Smithtown, an exploratory environment for learning principles of microeconomics. The ITS was shown to teach students in less than half the time, yet they learned the material as well as those students receiving traditional classroom instruction.

The importance of adapting to student differences is shown by the field called Aptitude-Treatment Interactions (ATI). ATI studies validate that different students will learn better if the teaching style is appropriate to their particular aptitudes. Conversely, students will perform poorly when exposed to methods that conflict with their aptitudes [Shute 1992].

3.0 Phase I Objectives/Accomplishments/Results

During Phase I, we have completed all 6 Phase I technical objectives described in the Phase I proposal and reproduced below, along with an explanation of the effort that occurred. A more detailed description of the Phase I effort follows this list.

1. Identified Air Force Training Requirements. Working closely with the Air Force, we identified specific domains for which training must foster cognitive flexibility in the student. Those domains included AWACS, Air Campaign Planning, Aimpoint Selection for laser guided munitions, and Paramedics. The Air Force also has a requirement to reduce travel costs related to training. Internet delivery of training is meant to address this issue. Furthermore, we determined the most pressing training requirements related to AWACS tasks.

2. Developed Strategies for Student Model Development. We investigated strategies for intelligently using examples to help students develop not only familiarity with course materials but also mental models that reflect a true understanding of the domain. This research was conducted within several domains - AWACS Weapon Director (WD) tasks, AWACS Surveillance tasks, and high school / college math and science. Some types of tasks are described as "more of an art than science," usually because it is hard to enumerate principles. These tasks tend to require the use of more examples and a wide breadth of examples but relatively little explanation for each example. The opposite extreme tends to have very specific principles which must be learned and applied. These tend to require fewer total scenarios (though several for each principle) but more explanation as to why specific steps were taken, which references the principles.

3. Assessed the feasibility of Semi-Automated Elicitation of Domain Knowledge. To facilitate course authoring, we explored the feasibility of representing domain knowledge as an organized collection of cases rather than as an explicit expert mental model or expert system and the ability of domain experts to directly input into this representation without requiring significant computer training. We developed an organization for these cases and have represented cases in it. Domain experts have seen the cases and can readily understand the content. We have implemented the case structure in software.

4. Developed Strategies for Mental Model Assessment. Through automatic intelligent indexing of exercises and intelligent comparison methods, we developed general analytical routines for assessing a student's mental model. For example, the ITS prototype can assess an AWACS WD's knowledge of, and ability to apply principles of radio communication, while performing in tactical scenarios.

5. Case-Based Representation and Reasoning Architecture. We produced a generic architecture for the case-based Internet ITS, which is described in Section 4.2.1. This was prototyped in Phase I and demonstrated the benefits of CBR for both automated knowledge acquisition and intelligent, student-responsive tutoring.

6. Prototype Development. We developed a proof-of-concept Internet prototype accessible with common Internet browsers based on the system architecture. The prototype demonstrated important CBR functionality both as an instructional design system and an Internet ITS implementation within a subset of the AWACS training domain. Initial versions of the authoring prototype and AWACS ITS are complete and were demonstrated in early December.

The first result from the Phase I effort was the concept of the "loose confederation of ITSs" idea, described in Section 4.1.3, Internet Delivery of IITSAT and ITSs, and a design for its implementation is described in Section 4.2.3, Internet ITS Server. The inspiration for this original idea came from the world wide web and the way it is currently used. People browse one web page, which may have links to other web pages, which they may follow and visit, even though the authors of the separate pages are probably unknown to each other. Further, search engines exist to help guide end-users to pages which may be of interest. Our concept for ITSs follows naturally from this model.

In Phase I, we also designed and implemented a scheme which would allow the required communication between ITSs. In our concept, a central ITS server keeps a master list of all ITSs, their capabilities, and a global list of principles. The ITS authoring tool running locally on the author's machine must be able to both get information from this server, and transfer information back. Further, once the ITS is fielded, the author needs to be able to send course updates and an instructor needs to be able to get the student models. Finally, the cooperative ITSs must be able to send student models to each, launch each other, and get information about each other. We also wanted to maintain the flexibility for the ITSs to be either running on a local machine or as a Web page. We implemented the required communication capability and demonstrated how a local ITS or web-based ITS could launch and communicate with Web-based ITSs or the central ITS server, automatically getting or giving the required information. These communication capabilities were implemented in three forms, to maintain flexibility - as CGI scripts (executable from Java Scripts), as a DLL (executable from local PC), and as a standalone executable. Thus we proved the feasibility of this concept by implementing the required communication functionality in Phase I.

We developed and implemented a very general structure to represent cases, applicable across several domains. That structure represents examples, exercises, and scenarios for instructional purposes in a common format. Each includes a multi-media description of the problem, which may evolve over time (as in tactical scenarios); a description of the correct actions to take, which may include order-independent, optional, and alternative steps; multi-media explanations of why these steps are correct; the list of methods which determine whether the steps have been correctly executed by the student, and the list of principles required to know the correct action to take. This structure is described in Section 4.2.

We developed a general ITS Architecture and implemented a simplified version of it in Phase I. This architecture addresses aspects of instruction such as assessment, remediation, use of examples, and style of course presentation, and is described in Section 4.2. The Phase I prototype monitors student actions in scenarios, derives a model of the student's knowledge, selects the most appropriate examples for instruction and exercises for testing, and selects the Topics that address the areas that the student doesn't understand. It also provides a facility for the student to browse those topics himself. Based on the knowledge requirements of ITSs represented in this architecture, we designed and implemented a simple version of the ITS authoring tool, described in Section 4.2. This authoring tool allows non-programming instructors to author ITSs that are based on the CBR paradigm, that is, example-based instruction. The Phase I prototype allows instructors to enter scenarios, including a description, the correct steps (which may be order independent, optional, or alternative and which may take one of several forms), explanations, principles used, and assessment methods. The author can create a Principle Hierarchy and a Topic organization which can include multi-media, alternative descriptions, etc.

We held discussions with several members of the AWACS community to determine their most pressing training requirements. The number one concern of all AWACS WDs is Radio Transmission (RT), that is, communication with other aircraft, especially fighter pilots. This was described as 50% of the job, something that usually needs to be improved on, something that practice would help considerably, and the number one reason why some weapons directors aren't as good as others or even have to be dropped.

To address this most pressing concern in Phase I, we implemented the AWACS WD communication ITS, using the prototype authoring tool. Scenario descriptions consist of animations of the WD radar console as well as recordings of fighter pilots making requests, reporting tracks, or sorting targets among themselves. Scenarios also include the expected WD communication alternatives at various times, an explanation as to why those are correct, the list of principles that the WD must understand to know whether and how to say the communication, and the list of methods that will be used to determine if it is acceptable. The types of principles that are being tested include knowledge of enemy tactical formations, friendly fighter target sorting criteria, required communication syntax and brevity words, the ability to be concise, and when to say how much so as to give the fighters the crucial information they need without overwhelming them with information that they don't need immediately.

Based on our experience with developing the AWACS ITS prototype, we refined the authoring tool prototype. In December, we demonstrated the authoring tool and AWACS ITS to Armstrong Laboratory.

4.0 Phase II IITSAT

4.1 System Functionality

Recent advances in Internet technology provide the unique opportunity to distribute Air Force training across multiple sites and dramatically reduce travel-related training costs. Not only can students receive training at their own local sites, but instructors can monitor students' progress from a distance, and course authors can maintain and update training material across the Internet.

The use of an intelligent tutoring system (ITS) for individual or team training will achieve many of the same benefits as one-on-one instruction. Additionally, an ITS can promote instructor productivity, cope with the higher skill requirements for trainees, provide tailored instruction and remediation, while allowing flexibility in teaching methods. When ITSs are enhanced with interactive multi-media, students demonstrate greater motivation during training and greater retention of course material. Thus, a multi-media Internet ITS provides instruction that is at once cost-effective and potentially superior to traditional classroom teaching methods, and it can be delivered across a great distance with substantial reduction in travel costs.

Internet ITS Use

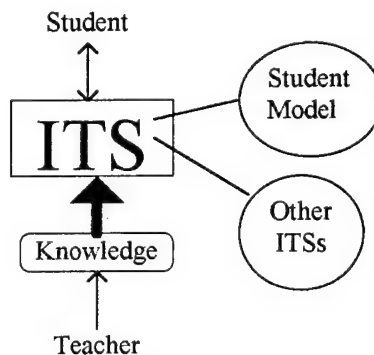


Figure 1, Intelligent Tutoring System Context

Pictured above, in Figure 1, is the ITS in context. The ITS interacts with the student, monitoring his actions and the information that he is reviewing. Based on this monitoring, the ITS maintains a model of the student's knowledge. It uses this model, along with pedagogical knowledge and student input, to decide what the next type of instruction should be. It might be to display an example relevant to the principles being taught at the time. It may be to test the student with an exercise, based on his student model, using principles that the ITS believes he may have learned. It may be to debrief the student on the mistakes of his last exercise. Or it may be to formulate a remedial course of instruction, based on the deficiencies in the student's current model. These remediations may take the form of examples or general topic information.

Finally, the ITS may decide that the deficiencies the student possesses are best addressed by a different ITS. This might occur if prerequisite knowledge appears to be lacking. For example, the prerequisite for a high school physics course might be algebra. If the ITS determines that the student is weak in this area, it might send him to an Algebra tutor on the Internet. It would thus need to have some knowledge of other ITSs and their subject matter and capabilities. SHAI developed and prototyped such a scheme in Phase I, which is described in more detail in Section 4.1.3

4.1.1 Intelligent Tutoring Systems Overview

An ITS can capture much of the expertise of the human instructor and deliver it either in a default structured format or through student-driven access capabilities similar to the one-on-one student-teacher interaction. Additionally, an ITS can monitor an individual student's progress and tailor the specificity and pace of further instruction accordingly, which can be an improvement over the traditional teacher-textbook combination which relies on a static printed text. An ITS is also extremely effective at managing situational examples and generating or retrieving new examples based on diagnoses of the student's mental model. Our Phase I prototype clearly showed the ability of an ITS to tutor students effectively in the AWACS domains. Furthermore, several of our previous projects (see Related Work) have proven the ability of ITSs to train students in other complex domains.

Expert Model

ITSs have shown great promise in military training domains. However, the key to the effectiveness of this instruction is the technical knowledge the system contains. Traditional approaches to development of ITSs are hampered by the knowledge acquisition bottleneck - the need to construct an explicit expert mental model. We propose an innovative approach to Internet ITS development using a case-based reasoning (CBR) paradigm. A CBR approach largely circumvents the difficult and time-consuming process of constructing an explicit expert mental model. We do not need to develop an expert system that exactly models the domain expert and exhibits identical expert behavior. Nor do we need to precompile a body of expert knowledge which anticipates all student interactions and errors. Instead, one or more experts' knowledge is contained in the examples. The expert is asked to present a problem and its solution, with an explanation of the steps required to produce the solution. The explanation refers to principles or concepts underlying the example. Thus, the principles serve as the organizational structure of the knowledge, and the examples serve to illustrate concepts. Because a case-by-case approach to knowledge gathering is highly intuitive, authoring of the training course is greatly simplified and requires no special computer training. Further, maintenance of the Internet ITS and addition of updated course material is accomplished primarily through addition of new cases. Our Phase I prototype authoring tool clearly demonstrated the feasibility of providing an ITS authoring capability to instructors.

Student Model

Typical Computer Based Training (CBT) systems try to assess student knowledge through the standard question and answer format. This is unrealistic for most domains and primarily tests the student's recall, memorization, and reading comprehension. It also misses the opportunity afforded by the use of a computer system since it merely mimics in computer form the traditional paper and pencil test. In most domains, the important assessment issue is whether the cognitive principles presented to the student can be applied by the student in realistic tasks in operational settings. By using a scenario, or case-based approach, actual problems and tasks that the student will face in an operational setting are used for assessment. Scenarios are also used for instruction, thus demonstrating how principles are applied, instead of just describing what they are.

A case-based approach offers the further benefit of automatically or semi-automatically generating the student's mental model. The only completely accurate way to assess a student's mental model of a given domain is to combine performance records from specific scenarios with responses to specific cross examination, which consists of questions designed to get a direct explanation of the motivations for the student's actions and reactions in a given situation. In other approaches, this knowledge is then precompiled in an attempt to anticipate all the mistakes that a student might make. However, this kind of analysis can be extremely cumbersome, both for the student and the instructor, so it is a worthwhile goal to automate this procedure by developing an analytical technique for inferring the structure of the student's mental model from performance records. Then, active questioning of the student is used simply to verify his understanding.

There are a number of obstacles to the automation of the analysis of the student model. First of all, a student may accidentally select the correct procedures without having a thorough understanding of the domain. Likewise, a student may act incorrectly due to one of several shortcomings in his mental model of the domain. In either case, it would seem that direct questioning would be the only way to determine whether the student's performance reflects true understanding. Secondly, the student may have developed a skill with the procedure in the testing environment but not have sufficient understanding of the overall domain to be able to apply learned concepts in new situations.

The intelligent use of examples is a potential way to counter these difficulties with the assessment of a student mental model. By presenting exercises requiring knowledge of principles and counter principles, it is more likely that correct student performance reflects true understanding. For example, in the domain of detection avoidance, it may be better in one situation to fly over land, and better to fly over water in another. If the student truly understands the motivations for flight path decision making, he would choose correctly in both cases. Otherwise, if he always simply chooses to fly over land, he may accidentally select the correct choice in the first case but will select the incorrect choice in the second case, thus indicating the problem with his mental model. If examples are automatically indexed by the various aspects of a domain representation that they correspond to, they

can be used to identify these kinds of shortcomings in the students' understanding. The Phase I prototype demonstrated that scenarios could be used to derive a student model of the trainee.

Furthermore, the more examples presented to the student, the more likely it is that the student will be able to apply concepts in new situations without linking parts of his mental model to the specific contexts of training examples. If these examples are not mere text, but sophisticated multi-media scenarios, simulations, animations, 3-dimensional graphics, audio, video or hypertext, they will offer greater realism for the student, engage his interest more fully, and ultimately result in superior training and retention of concepts.

Success of the CBR Approach

Stottler Henke Associates, Inc. (SHAI) is a leader in the field of CBR. Additionally, we are confident of the generality of this unique approach to Internet ITS design because we have used it with much success in many training domains, including AWACS Weapons Directors (the Phase I of this effort), Surface Warfare tactics, AEGIS weapons doctrine, dismounted infantry tactics, paramedic techniques, environmental design, sonar analysis, and AEGIS team training. In the Phase I research effort, we proved the feasibility of these techniques through development of a proof-of-concept Internet ITS authoring tool and application of the tool for AWACS Weapon Directors.

The validity of the case-based approach has both intuitive appeal and empirical backing. As early as 1940, [Gragg 1940] argued for case-based instruction. By presenting (often with a simulation) cases which illustrate the important principles, the student can see how principles are applied in operational contexts and tasks. It also defeats the well-known problem of inert knowledge first described by [Whitehead 1929] and frequently validated by other researchers. Inert knowledge is information or principles that a subject knows and can recall, but which he does not apply when the situation clearly calls for it. Case-Based instruction (and related concepts such as anchored instruction, scenario-based instruction, simulation-based instruction, and situated instruction) overcome this problem by showing students the application of principles in an operational setting and forcing them to apply them as well.

Case-Based instruction embodies other relevant theories of instruction. For example, research has shown [Farquhar, et al. 1992] that in dynamic environments, the provision of graphical dynamic simulations improves the development of a proper mental model in students. Thus, the use of scenarios presented graphically and dynamically is important to illustrate related principles. Another theory of instruction is situated-learning, where the importance of tying learned knowledge to tasks in an operational environment is emphasized. Again this supports the use of a simulation that approximates the operational environment for which the student is being trained.

We have completed a number of ITS projects for the military and, clearly, military instructors and trainers favor the use of tactical simulations and believe that the best trained officer will be the one who has experienced the most tactical scenarios. We have also worked on projects with Dr. Beverly Woolf. Her research also indicates the benefits of scenarios played in simulations for instruction and assessment.

[Towne et al 1983] describes one study where one group of Navy technicians used a simulation-based trainer to practice fault diagnosis while another practiced using the real equipment. All students then took a trouble shooting test on the real equipment. No significant differences were found between the ability of the two groups.

4.1.2 ITS Authoring Tool Overview

An ITS Authoring Tool is designed to allow a course instructor to easily enter domain and other knowledge without requiring computer programming skills. The Internet ITS Authoring Tool (IITSAT) then generates an ITS for the particular course. The IITSAT must facilitate the entry of examples/exercises, including problem description, solution, solution steps, and explanations. The examples may be in the form of scenarios or simulations. It must allow entry of the course principles in an organized fashion. It must be able to integrate Internet multi-media courseware (developed with well-known authoring tools) especially when that courseware includes descriptions of the principles or provides motivational passages. In addition to course knowledge, the instructor can enter other domain knowledge, pedagogical knowledge (how best to teach), and student modeling knowledge (how to assess actions and how to determine mastery). Defaults would typically be provided for these types of inputs.

Our IITSAT will allow easy development and maintenance of training systems which are based on pedagogically sound instructional strategies, validated through empirical research. The case-based approach, using scenarios and simulations to teach as well as assess student knowledge, is such an approach. The training systems created with the tool will include a variety of media, including graphics, Java scripts, HTML, video, sound, animation, and simulation.

4.1.3 Internet Delivery of IITSAT and ITSs

The Internet provides an opportunity to reach more personnel in more geographically diverse locations by providing long-distance learning and delivery of multi-media. Unfortunately, current Internet training systems have been developed in a variety of styles with little reference to sound pedagogical principles. Authoring tools based on theories of learning and instruction are needed to facilitate development and maintenance of interactive, multi-media ITSs for use across the Internet. Most Internet training systems are simple page turners. They often have an impressive array of graphics and sometimes video and animations, but the only customization to individual students is accomplished by having the student self-pace the presentation of the information, sometimes via navigation through a hyperlinked, multi-media text book. More advanced

systems are needed which can assess student knowledge, derive a student model, diagnose deficiencies, and provide a tailored remedial course of instruction. Such a system would off-load many of the tasks traditionally performed by instructors.

Once Phase II is complete and instructors can easily author Internet ITSs, a previously unavailable opportunity is presented. Since the instruction is now in an automated, Internet-accessible form, ITSs developed with the authoring tool can coordinate their efforts in the training of a particular individual. These ITSs can be maintained by separate instructors who have little or no knowledge of each other and are located in geographically distributed areas. In Phase I, we devised and prototyped a scheme which allows the ITSs to notify each other of their existence and training capabilities. The scheme also enables one ITS to make specific training requests of another, remote ITS, passing it the trainee's current student model and training needs. This "loose confederation of ITSs" concept is a completely new paradigm for ITSs and education which could revolutionize training. ITS authors don't have to duplicate each other's work (as in overlapping or prerequisite material) but can take advantage of it. Instead of developing an ITS containing all related and prerequisite knowledge, course authors only have to author systems in their primary field of interest and let others take care of related fields. Every trainee has equal access to all ITSs and can make use of the best ones in each field without having to know about them explicitly because the system guides them there. The ITSs have universal access to students. That is, an ITS author knows his ITS will get used without having to promote its existence. Authors can concentrate their time on what they want to teach, not on other subjects, and not on access and promotional concerns. Of course, the privacy of each student's mental model needs to be addressed. The central server can keep them in encrypted form, and students can maintain their own passwords to access their own student models. This would simply require that each student enter a password each time he begins a new ITS session.

SHAI has significant ITS, ITS Authoring, and multi-media development experience. In addition, SHAI has experience developing Internet applications. These applications typically include Java language programs, HTML files, C++ code called from Java software, and multi-media files. The applications can span the full gamut from being entirely hosted on a Web server and accessed from a remote location to being entirely downloaded from the server and being hosted on the remote machine to hybrid systems which include code executing both on the host and remote machines, communicating appropriate information such as updated courseware, student models, or new training objectives.

4.2 Design

This section is divided into four subsections. The first describes the general ITS architecture. The second describes the ITS authoring tool. The third describes the Internet ITS Server. The fourth details the AWACS ITS which will be developed with the authoring tool, including an example of how an instructor would use the authoring tool to create the AWACS WD ITS.

4.2.1 General ITS Architecture

Pictured below is the general ITS architecture, where the gray boxes represent knowledge which must be entered by the instructor (using IITSAT) to create the ITS. Because students often learn best by example, the tutoring system is designed around example problems. The examples are made more vivid by interactive multi-media technology. An example consists of a problem description, solution, and explanation or steps leading to the solution. An exercise is extracted from an example by only showing the student the problem description. He must then generate the solution himself. His solution and solution steps can be compared to that of the exercise for grading, immediate feedback, and remediation.

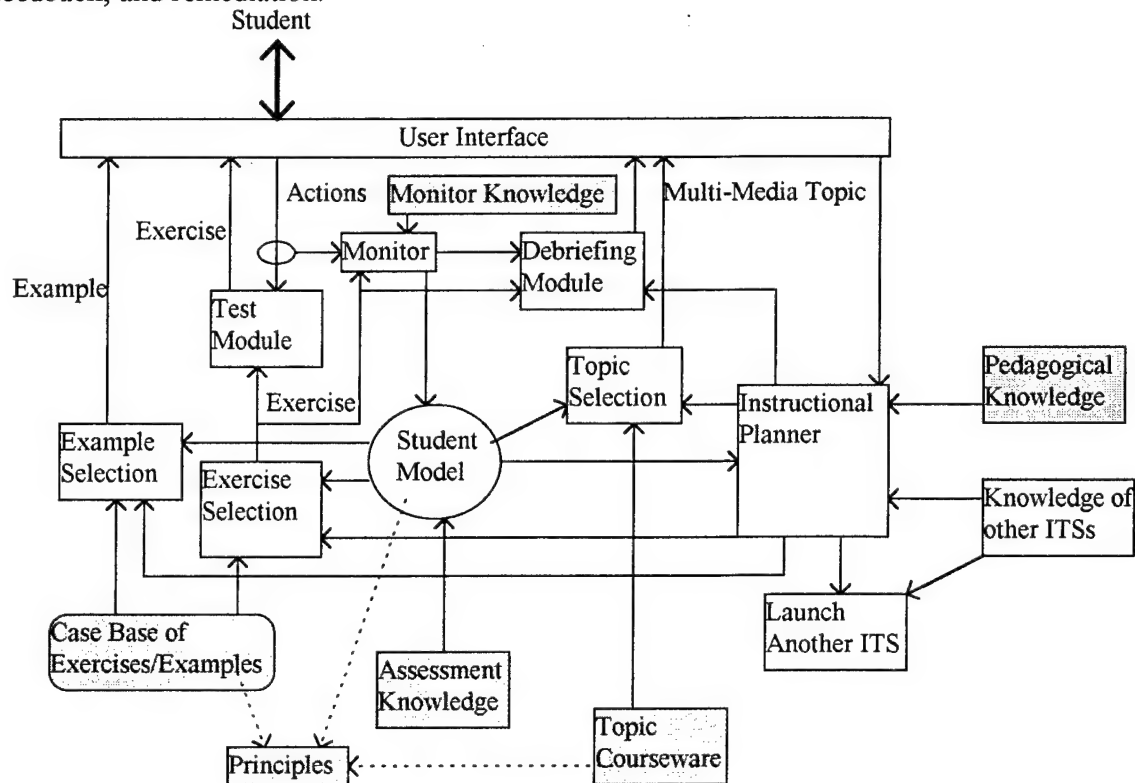


Figure 2, Phase II ITS Architecture

The instructor can specify the level of expertise of each example (whether it is appropriate at the beginning, middle, or end of the course) or could specify the ordering of principles and problem-solving methods. This would constrain the presentation of examples implicitly based on the principles and methods referenced in the explanation. For example, an example whose explanation contains a reference to a principle not taught until the end of the course cannot be presented until the end of the course.

The ITS could teach by first presenting an example and its solution, showing the steps that were used to arrive at the solution. Principles introduced for the first time would be followed by their detailed explanations. In order to tailor the course of study to

the individual student, we will keep a model of each student who uses the ITS. The student model will contain the student's actions and decisions during different exercises, the principles, procedures, and techniques which have been presented, and those that have been mastered based on performance of exercises. The set of principles, procedures, rules, and tools referenced in the solutions of problems the student has solved successfully represent the student's acquired skills. Based on the pattern of his unsatisfactory performance on exercises, a set of topics, principles, or combinations of them, can be developed which form an hypothesis as to what information the student does not understand. Based on this hypothesis and the exercises solved incorrectly, similar examples can be shown to increase the student's understanding. It may be appropriate to force the student to re-experience some of the course topics as well, perhaps to a greater level of detail. Based on this hypothesis, a new set of exercises can be generated for testing the success of the remediation. The student model can also be referenced by the course instructor to monitor the student's progress through topics and his performance on the problems. The student model will reflect the skills, knowledge, and error-rate of the student. The model will evolve in size and complexity as the skills and knowledge of the student increase.

In designing an ITS, it is important to maintain flexibility as to how an example should be presented. An ITS must be able to accommodate alternate teaching strategies and evolutions in course material or mental processes. Further, it must present the example as animated simulations, high resolution pictures, annotated video footage, hypertext, as well as more standard forms of text and graphics.

Since the Exercises and Examples reference the Principles of the domain which they test and/or illustrate, Figure 2 shows a dotted line from them to the Principles. Similarly, the Student Model represents the student knowledge by reference to these principles, based on what actions the student got right and wrong in the exercise. There are actually two different representations for each Principle. One representation is kept at the ITS Server (see Section 4.2.3), so that it knows what each ITS teaches. The second one is a local copy kept with the ITS. The local copy represents the terminology used by the local ITS for the principle. In this way, different ITSs can use different terminology but still communicate via a translation step that occurs at the ITS server.

Determining which actions were right and wrong is performed by the Monitor module. This is one of the most difficult steps. In traditional ITSs, an expert system is applied to the same problem as the student, and their actions are compared. Unfortunately the creation of the expert system is difficult or impossible even for a skilled knowledge engineer in most domains. The task would be hopeless for a non-programmer, such as an instructor. In our approach, the correct actions are stored with the exercise, so the Monitor need only compare the student's actions to these. This comparison may need to take into account several aspects of the domain, such as order independence of the actions. The architecture shows that the instructor could enter knowledge of how to assess correctness of actions (Monitor Knowledge).

The Exercise Selection module retrieves the best exercise based on a number of factors. First, it considers the primary objective of the test, i.e. the set of principles that the ITS is most interested in seeing if the student knows and can apply. It then tries to retrieve the exercise which contains as many of these as possible, while not containing any principles that the system knows the student does not know. Similarly, the Example Selection Module attempts to retrieve the example which includes the most principles that the student needs to understand. As a leader in CBR retrieval, SHAI has developed several such retrieval modules.

The decision as to what the next appropriate instructional event will be is decided by the Instructional Planner. One instructional sequence is described above. In other domains, or for other students, an appropriate sequence may consist of "exercise, debrief, remediate with Topics, and repeat," where the exercises selected depend on the student model. This would be appropriate in situations where it is expected that the student is already familiar with the material, perhaps having already read a text book. In another situation, the most appropriate sequence might be "motivational example, Topics, examples, exercise, debrief, remediate with examples and more detailed topic information, and repeat." This would be appropriate when the student is completely unfamiliar with the material and needs to access the multi-media textbook (Topics) before being expected to perform. From the student model, the Instructional Planner might determine, in another situation, that an exploratory discovery environment is most appropriate. To determine the next event, the Instructional planner considers both the Student Model and Pedagogical Knowledge, entered by the instructor, which describes the preferred way to teach. This knowledge might specify a fairly rigorous pattern of events, or one which adapts itself closely to the student model, or one that allows the student considerable freedom in choosing for himself the next event.

When the Instructional Planner has determined the need to show Topics to the student, the appropriate one will be selected, based on the Principles each describes. There may be times where essentially the same principles are described by different Topics. Two possibilities are that essentially the same information is present in different forms, such as a description being primarily text, audio, or graphical, or, that the principle is described in different ways, such as a description in one and an historic example in the other. By tracking the student's performance on exercises after receiving Topic information (either for the first time or for remediation), we can determine how well the student was able to learn from the particular Topic used. The system can then begin to learn which types of Topics the student learns from best, i.e. his learning style. SHAI has already developed this general capability and applied it in several domains.

One of the most effective instructional techniques, and one which an ITS is perhaps best suited to address, is immediate feedback. As the Monitor Module is determining the correctness of the student's actions, it can send the results to the Debriefing module which is assembling a debriefing, based on the student's actions, the correct actions as described in the exercise, and the rationale as to why those actions are correct, also stored with the exercise. Immediately upon completion of the exercise (or

during), this debriefing can be presented to the student. SHAI has developed debriefing capabilities in several specific ITS projects.

After the Monitor Module has determined which *actions* the student got correct and incorrect, the Student Modeling facility can determine, by examining the principles attached to the correct actions in the exercise, which *Principles* the Student has gotten right and wrong. It then must decide which Principles the student has mastered. Is performing one correct action involving the Principle enough? Three in a row? 80% correct? Should our belief in the student's mastery decay over time if he has not been tested on it for a long while? These types of questions are answered by the instructor in the Assessment Knowledge module. Further, the Student Model must not only keep track of which principles the student has mastered but also which ones he needs testing on (ones he might know), which ones he has failed in tests (ones he doesn't know), and which ones he has seen no information on at all (ones unheard of).

A final element of the ITS, should be knowledge of other ITSs which relate to this ITS's subject matter. As described in Section 3.1.3, this knowledge can be maintained automatically by the ITS system so that individual ITS authors do not need to know explicitly of each other's ITSs. The system will automatically know of the related ITSs and launch them, for the student, with the student's Student Model, when appropriate.

This project seizes an opportunity to greatly increase teaching effectiveness by using experience encapsulated in examples for tutoring. The ITS will retrieve tutoring examples and problems whose explanations reference the principles required at a particular time by a particular student's needs. This will allow the student the opportunity to receive a tailored course of instruction. The general ITS design framework developed for this project will not only result in an implementation in a particular domain but also a reduction in development time for future ITSs in other domains.

A computer-based ITS is an excellent tool for presenting examples because many complex situations can be described and enacted through simulations that allow the student to not only have a better mental model of the situation, but also to observe the results of specific actions. Within a simulation, a student can either be shown the appropriate steps for a given situation, or required to perform them himself for testing purposes. The interactivity of simulations makes them superior to a textbook or pencil-and-paper testing approach, because they facilitate the student's involvement in the described situation. This leads to a more accurate measure of the student's flexibility in applying concepts differently for different situations. CBR is used to represent the examples and decide the most appropriate examples or exercises to present and when to present them.

An interface for entering examples or exercises as well as course information is a useful feature not only for reusing the ITS structure in a new domain but also for expanding its information base in a given domain. This kind of interface consists of a basic prompting mechanism by which a course designer can enter new exercises and the

associated course information. Examples are indexed by the components of the domain that they correspond to, which will be treated as principles for teaching purposes in the ITS. Each new example may refer to principles that are already in the ITS or to new principles, in which case the designer can be prompted for the information associated with these new principles. Principle information may include course materials in any of a variety of media and any relevant relationships with other principles (hierarchical, conflict, etc.). With an automated course design utility, the ITS can elicit all the appropriate indices that contribute to an organization for quick student access and intelligent tailored instruction. This is discussed more in the following section.

4.2.2 Internet ITS Authoring Tool (IITSAT)

Shown below is the architecture for IITSAT, where the gray boxes represent the knowledge that the ITS author must enter. These are basically the same as the gray boxes shown in Figure 2, the Phase II ITS Architecture. Each element is described below.

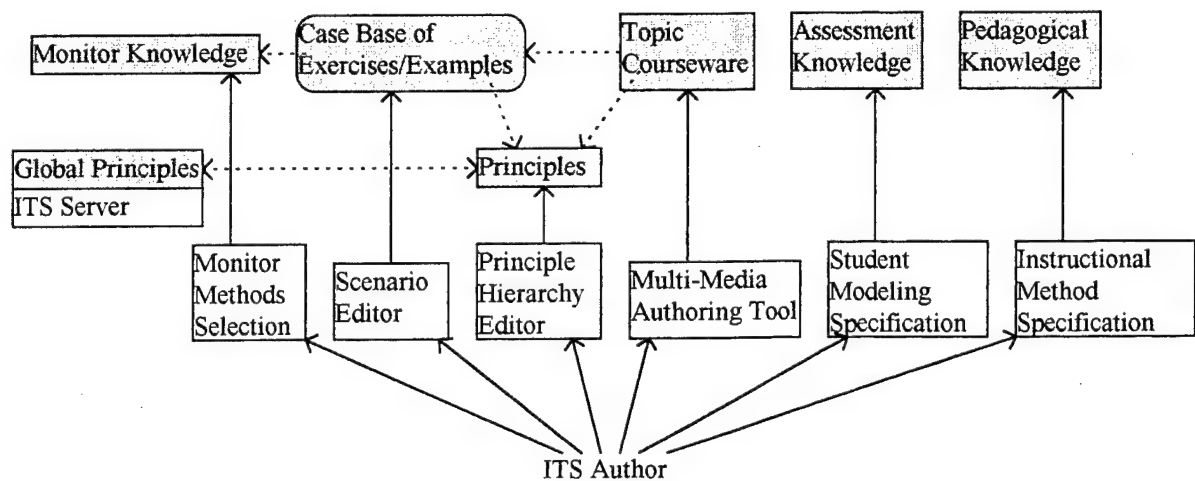


Figure 3, Internet ITS Authoring Tool

Case Base of Exercises/Examples

Perhaps the most central knowledge that the author must enter to create an ITS is the case base of examples and exercises. Each case or scenario includes a multi-media description of the problem, which may evolve over time (as in tactical scenarios); a description of the correct actions to take, which may include order-independent, optional, and alternative steps; multi-media explanations of why these steps are correct; the list of methods which determine whether the steps have been correctly executed by the student; and the list of principles required to know the correct action to take. The description should typically consist of multiple types of media, some of which may be somewhat domain dependent, requiring considerable flexibility to be built in. In our design,

descriptions can be hierarchical groups of materials where any material is a file name and an indication of what software is required to display the material. For example, the file might include the information required to run a tactical simulation and the software listed is the simulation executable which is called with the file to display the simulated tactical scenario. Thus, the Scenario Editor will allow for the various aspects of the description to be entered in various ways, including the use of tools external to the authoring tool (such as a scenario editor for a tactical simulation).

The correct actions to take are represented as a series of possibly hierarchical steps. Steps can be grouped where the groups correspond to steps that must all be executed in a specific order, must be executed in any order, or only one of which needs to be executed. Furthermore, each step may be broken down further into its own groups of steps. Each step may also occur in its own (domain-specific) tool, if the tool has been interfaced to IITSAT. Most tools primarily represent their results as files. Interfaces for these tools merely consists of the ability to compare their files to those created by the instructor. Thus if the problem solution requires actions to occur in several different software packages, this is supported. Examples include the user interface to a tactical simulation, communications, calculator, drawing package, notepad, equation solver, etc. The Scenario editor will allow the entry of the step structure, including entry of which tool should be used for that step.

Attached to each step is a multi-media description which explains why the step is correct. The representation structure is similar to the problem description, though one feature will only tend to be used here. The material object described above is actually a list of files and media types. In the case of explanations, there may be multiple files that essentially say the same thing but in a different format (audio, text, graphical, etc.). By monitoring the student's performance after receiving instruction, we can develop a profile of his learning style. For example, if he fails a specific step, and in the debriefing is given a particular explanation, and then if he fails a similar step in a similar scenario (one that requires the same principles), the system can surmise that the media was not well-suited for this student. SHAI has already developed this capability in a previous project.

Monitor Knowledge

Each step includes the list of methods which should be executed to compare the student's actions to the correct ones listed in the scenario by the ITS author. In the diagram above, this is called Monitor Knowledge. The simplest method would be one that compares the actions to see whether they are exactly the same. Actions may include parameters and ranges of these parameters may all be considered correct. Some actions relate to communication and it may be more important that all aspects of the communication are present, rather than they appear in a certain order. The authoring tool will support several user-selectable methods for the comparison and include the ability to add others.

Principles

The Principles are the aspects of the domain that the student needs to learn and apply when solving problems. We use the term, "Principles," very loosely. It includes domain principles, as well as facts, heuristics, strategies, tactics, procedures, etc. The Principles are entered through the Principle Hierarchy Editor, which allows the ITS author to maintain a reasonable organization. As shown above, the local set of principles is kept consistent with the global set which resides on the ITS server. This allows ITSs to cooperate. Typically when the author starts defining the principles for his ITS, he is first encouraged to check and see if some other author has already defined them, perhaps with different terminology. If so, he is encouraged to use his own terminology for the principle but link it to the global principle. In this way the ITS can convert terminology from one ITS to another, authored by a different individual. Each Principle, in addition to storing the local and global name, includes several alternative descriptions in a variety of media. These alternate descriptions are also varied and include descriptions, examples, historic examples, exaggerated examples, etc. These are used, as above to tailor instruction to the student's preferred learning style. Principles also include lists of other principles which indicate which ones are required prerequisites or suggested prerequisites. These may be taught by this ITS or another one.

Topic Courseware

From our perspective, the most important goal of each Topic in the Topic Courseware is to explain a Principle or group of related Principles. Thus, when the ITS determines that a student is having trouble with a principle or set of principles, it sends him to the appropriate Topic. Each topic therefore includes the list of principles it teaches. Each topic also includes a list of several topic elements. Each element includes a type which indicates what type of instructional instrument it is - description, example, exercise, test, etc. The descriptions are likely to be in a multi-media format. Topics can be organized into a hierarchy and presented like an electronic textbook. In fact, a popular multi-media CBT authoring tool may be used to develop the Topic Courseware. This might be considered an electronic textbook companion to the ITS, although it will have the capability to display specific Topics, when required for the student.

Assessment Knowledge

The instructor should enter the criteria for mastery of a Principle. After the Monitor Module has determined which actions the student got correct and incorrect, the Student Modeling facility will determine, by examining the principles attached to the correct actions in the exercise, which Principles the Student has gotten right and wrong. Based on the history of correct and incorrect actions relating to a particular principle, the ITS will use the instructor-entered mastery criteria to decide if the principle has been mastered. Typically, the ITS author would select one of a few different methods and fill in the particular parameters, such as the number of correct applications in a row, or the percentage correct and how many of the last applications to include in the calculation.

The author may also want to include a decay factor so that after enough time has passed since an application of a particular principle, the student's mastery is in doubt, and he will be retested. Further the Student Model must not only keep track of which principles the student has mastered, but which ones he needs testing on (ones he might know), which ones he has failed (ones he doesn't know), and which ones he has seen no information on at all (ones unheard of).

Pedagogical Knowledge

The instructor will have several different instructional methods to choose from for inclusion in the ITS. These are stored in the Pedagogical knowledge. These basically relate to two broad issues. First, who should control the sequence of instructional events - the student, the author, or the ITS? Second, especially in the case where the author controls the events, what should that sequence be? At one extreme, the student could have perfect freedom to browse the principles, topics, and cases, and request to be tested when he wants, on the cases of his choosing. All the while the ITS would track his progress and maintain the list of principles he has mastered. It could then tell him (or he could browse his own student model to find out) which principles he still needs to cover and be tested on. At another extreme, the ITS could decide which examples and topics to show the student (and in what media type), based on the principles he hasn't heard of and the ones he has failed. It could decide when and which cases to use for testing, based on the principles he might know. This latter extreme gives the ITS the most freedom to customize its instruction. Alternatively, the author could specify the order to show topics and for each topic, which exercise should be used to test mastery. If parts of this exercise were missed, the ITS needs to have the freedom to choose the remediation and follow-up exercises, since each student might miss different aspects of the exercise.

4.2.3 Internet ITS Server

One of the interesting and unique uses of the Internet is the way people browse the World Wide Web. They look at some interesting page; if they need more information, they jump to another page. Web pages are developed and maintained by a haphazard, loose confederation of individuals.

Now consider that same model for ITSs - a loose confederation of mini-ITSs maintained by separate individuals with little or no knowledge of each other's existence. A user interacts with one ITS. When it determines that he lacks knowledge in a related field, handled by another ITS, it sends him there, and so on. The user effectively browses through ITSs, but the ITS may be the one who decides where he should go to next. And when he switches from one ITS to another, his student model is sent to the other ITS so it instantly knows the state of his knowledge in related fields and his learning style preferences.

For example, consider four related ITSs - Wave-Theory, Sound Wave Theory, Electro-Magnetism, and Calculus. Suppose the student starts out interacting with the EM Tutor and it determines that the student is weak in Wave Theory. It sends him to the wave theory ITS, passing along his student model. While interacting with the wave theory ITS, the system determines he is weak in requisite calculus subjects, so it sends him to the calculus tutor, with his student model and a list of what the ITS needs to teach him (he only needs a portion of the calculus course as a prerequisite for the wave theory). After mastering the required calculus subjects, he returns to the Wave course and then back to the Electro-Magnetism course. The next subject he is assigned to learn is the Sound Wave course. His student model is passed to it, and it determines that he already knows a lot of wave theory (from the Electro-Magnetism course), so these topics can be skipped. Since he has also interacted with the Calculus tutor (since he needed to), anything not taught him while there, he's probably missing. If a few additional calculus topics are required, the ITS sends him back to the calculus tutor to focus on those topics. The calculus tutor itself might be a loose confederation of ITSs which each address different (or possibly overlapping) topics.

There are three capabilities delivered by the Internet functionality - ITS coordination as described above, transfer of the updated student model to the instructor's location, and transfer of updated ITS information from the author to the student's location. The design to implement these capabilities is shown below.

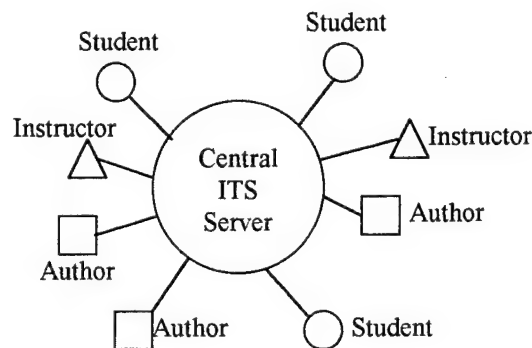


Figure 4, ITS Server

As shown, an author creates the ITS at one site, in consultation with the server and sends it the finished ITS (and any updates later). The ITS Server can thus keep a master list of all the ITSs and their capabilities. The student receives at his site the ITS (and updates) from the ITS Server, and his student model is transferred to the ITS Server. It can therefore keep a master list of all students. An instructor can access the student models of his students from his site. Since the local ITS that is running either as a web page or on the student's local machine was delivered from the ITS Server, it can automatically know of related ITS and when required, request these from the ITS Server to tutor the current student. The required communication capabilities were prototyped in Phase I.

In summary, the concept is a loose confederation of Internet ITSs (which know about each other) maintained by individuals with little or no interaction between them. These ITSs would tend to be maintained by people with an interest in education - teachers, university and college professors, and organizations with an interest. This idea is a completely new paradigm for ITSs and education which could revolutionize education. ITS Authors don't have to duplicate each other's work but can take advantage of it. Instead of developing ITSs with all related and prerequisite knowledge, authors only have to author systems in their primary field of interest, and let others take care of related fields. Every student has equal access to all ITSs and can make use of the best ones in each field, without having to know about them explicitly. The system guides them there. The ITSs authored have universal access to students. That is, if I author an ITS, I know it will get used without me having to promote its existence. (i.e. bi-directional universal education). Authors can concentrate their time on what they want to teach instead of other subjects or access and promotional concerns.

SHAI is already developing this concept and has developed the software to pass the needed information from ITS to ITS across the Internet, as well as to a central repository. The initial assumption is that all ITSs are authored with the same Internet ITS authoring tool, but this constraint can be relaxed later. This is so that they can pass knowledge of each other's existence around, so each one will know about the others and what they teach.

4.2.4 AWACS ITS

In Phase II, the IITSAT will be used to create an Internet ITS for AWACS Weapon Directors. Weapon Directors (WDs) have several different possible job assignments including Offensive Counter Air (Strike or OCA), Defensive Counter Air (DCA or CAP, which stands for Combat Air Patrol), Tanker Refueling, Check-in, and High Value Airborne Asset. Furthermore, each assignment has different phases, where different knowledge is required in each phase. The AWACS ITS will initially concentrate on communication skills, which includes knowledge of friendly and enemy tactics, situational awareness skills, the ability to be concise and credible, and knowledge of the formal syntax and vocabulary for radio transmissions.

The ITS author would probably first begin by entering scenarios which cover the various aspects of the domain that he wants to teach and assess. These scenarios would likely consist of radar screen animations, similar to what the WD would be viewing on-board the AWACS. Also included would be audio files for the different radio channels that he must monitor. Finally, the correct communications of the WD would be recorded by the instructor, possibly as an audio file which is then converted to text, using speech recognition software. This capability was demonstrated in Phase I. In the course of entering the correct actions, and the explanations as to why they are correct, he would begin to formulate the principles that the student must know to take each correct action. He would thus simultaneously begin developing the AWACS ITS Principle Hierarchy. He would also be shown the global hierarchy so he could see if either some of his principles

overlap with those in another ITS (not very likely in this case) or if some other ITS teaches prerequisite principles (which is more likely, since either fighter tactics or geometry are prerequisite knowledge which might be taught by a different ITS). The AWACS ITS principle hierarchy would be sent to the ITS server, for coordination with other ITSs.

As he developed the Principle Hierarchy, he would input multi-media descriptions of them. Perhaps these would be simple text or graphics, audio files entered with a microphone, or "talking head" videos. Or they might be entered with a multi-media authoring tool, such as IconAuthor. If desired, the instructor could even develop a multi-media text book to accompany the ITS in a tool such as IconAuthor, though this will probably not be done in this domain since it would duplicate the training that already occurs at Tinker AFB.

The Author would select the methods used to determine when the student's communications are correct. That method would most likely be one that checks if the student's radio transmission (RT) corresponds to one of a few alternative correct RTs. This check would ensure that the student's RT includes all the words of the correct RT (or their synonyms) and that it contains fewer than 20% extra words (meaning it is concise). Mastery for a principle would vary. The most critical principles would probably require 100% correct of the last 10 applications of that principle. The least critical might only require an 80% correctness ratio. The instructional technique entered by the author would probably consist of initial testing with a few scenarios designed to cover the broad range of RT situations. This would be followed by allowing the ITS to debrief and decide which remediations to apply including both examples and descriptions. Finally the ITS would decide which exercises to follow-up with. Thus, the instructional technique is initial testing followed by an exercise, debrief, remediate, and repeat loop.

When complete, the author would then instruct the authoring tool to create the ITS and send it to the ITS Server. Later, a WD in training downloads the AWACS ITS. He logs on for the first time and the ITS first checks the ITS server to see if his student model already exists. If so, it confirms that it is the correct individual and uses the existing global student model as a starting point. It first tests him with the initial scenarios. It determine that there are some parts of the Principle Hierarchy he knows well, some he does not seem to know at all, and some that he partly knows. One section that he is especially weak in is friendly tactics. He consistently misses communications whose cues relate to fighter pilots' discussions and the tactics that they employ. The ITS is aware that this knowledge, in addition to being taught in a cursory manner by the AWACS ITS, is taught in-depth by an ITS maintained by Mike Felhay (call sign "Guido") of the 963rd Tactics Shop at Tinker AFB (for example). It downloads that ITS, launches it, gives it the student's student model, and instructs it to teach friendly fighter tactics.

After addressing the student's most pressing concerns, the ITS can turn its attention to parts of the principle hierarchy that the student only partly knows. For example, the student appears to have mixed knowledge of the section of the principle

hierarchy relating to brevity words. The ITS chooses several scenarios to cover the gamut of brevity word usage to determine more exactly which ones he knows and which he doesn't. It remediates the ones he does not know, primarily through examples of their usage since his student model indicates that examples appear to fit his learning style in this part of the hierarchy. The ITS retests him after this remediation using different exercises which require their use. Daily, his student model is updated at the ITS Server for access by the WD instructor who will check his progress and perhaps engage in a specific communication to explain some principles that he is consistently missing. The ITS continues to loop through exercise, debrief, remediate (followed by an exercise again), until all WD principles are listed as mastered in the student model.

Meanwhile, the original course author (who may be the WD instructor) has been getting feedback on how the students are doing. It appears that almost all students have trouble with the concept of packages. He thus decides to add a Topic which describes it. The topic includes a motivational example which shows why the concept of packages was created (to indicate airborne tracks so far away from each other that they are unlikely to be coordinating their tactics). He then inputs the description, using a picture, which shows how far away tracks are when they are called "groups" and how far away to be called "packages". The topic also includes several correct and incorrect examples of the use of term "packages" and "groups." Finally he alters the instructional method, to present this topic before any exercise requiring the package concept. Thus, the author only needs to do additional work to put in extra detail on the principles that the students were having the most difficulty with. The author can tune and optimize his course over time, based on automatic student performance feedback.

He has also decided to teach additional principles. His ITS additions include the tanker refueling function (refueling and tanker intercepts), optimal selection and awareness of radio channels, and asset management. All of the ITS updates are sent to the ITS server and then sent on to the student sites, once each day when the student model is updated.

5.0 Phase I Prototype

5.1 Functionality

In general, the Phase I prototype implements all aspects of the full-scale system, though in a simple form and not necessarily integrated with each other. That is, the prototype includes representative ITS, ITS Authoring, and Internet functionality. Additionally, the prototype authoring tool was used to create a prototype AWACS ITS. The functionality of each is described below.

ITS

The Phase I ITS prototype presents a scenario to the student, monitors his actions in the scenario, compares those to an expert's description of the correct actions, then determines what principles the student applied correctly and incorrectly. It then debriefs the student, updates the student model with those lists and shows the student appropriate remediation. It then selects another scenario based on what principles the student needs testing on.

ITS Authoring

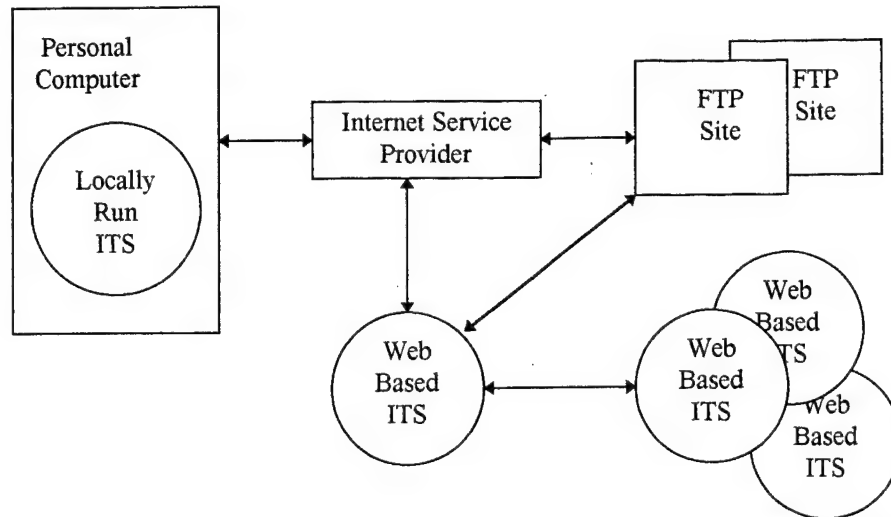
The authoring tool prototype allows users to enter the knowledge needed by the ITS. The prototype allows entry of the Principle hierarchy (an example is given on the next page), Topic structure, and scenarios. Topics may consist of files containing multi-media.

- AWACS Root Principle
 - Say it on Time
 - Be Concise
 - Don't be too wordy
 - Don't give unnecessary information
 - Don't give unnecessary fill-in information
 - Don't describe irrelevant groups
 - Brevity Words
 - Understanding the correct priorities
 - Hot groups are more important than cold ones
 - Close groups are more important than further ones
 - Describe big picture first then fill in information
 - Say important information first
 - Looking for Mistakes
 - Missing a Group
 - Sorting
 - Untargeted a Group
 - Understanding Enemy's Tactics
 - Champagne
 - Echelon
 - Ladder
 - ...

Local Name	Champagne
Global Name	Enemy Champagne Tactic
Description	<div>champain.txt TEXT DEFINITION champain.avi VIDEO EXAMPLE</div> <div>Add Edit Remove Display</div>

Internet

Using CGI scripting and JAVA programming, we implemented a way for an ITS running as a Web page on a Web server to pass information to another ITS running as a Web page on another Web server. We have also implemented a way for an ITS, running either locally on a PC with a current Internet connection or as a Web page on a Web server, to request and pass information back and forth to a central FTP site.



In Phase I, we used this communication capability to build prototype capabilities directed toward the three main required Internet functionalities - update of ITS coursework, transfer of updated student model for viewing by the instructor, and coordination of cooperating ITSs.

AWACS

Using the prototype we developed a prototype AWACS ITS, which concentrates on tutoring Weapon Director (WD) communication skills. A version of the Principle Hierarchy was shown in the Authoring section above. It tutors in the application of these principles by presenting AWACS WD scenarios in the AWACS simulation. These scenarios are presented as a simulated radar screen (reproduced on the next page) and simulated fighter pilot transmissions in audio form. The student responds to the simulation as he would on-board the AWACS - through verbal communication. His speech is processed with speaker independent voice recognition technology and compared to the expert's communications. Based on what communications he performs correctly and incorrectly, the system keeps track of which principles he can apply. It then debriefs him on the scenario and, if he continues to have trouble with a principle or set of principles, it remediates him appropriately. We added an additional, very small ITS, which tutors enemy tactics, so we can demonstrate how one ITS can make use of another. Finally, the AWACS ITS retrieves additional AWACS scenarios which utilize principles that the student still needs testing on or practice with.



5.2 Demo Sequence Overview

This section gives a brief overview of the Phase I prototype demonstration sequence. The detailed sequence, along with numerous screen dumps is given in the Appendix. The demonstration sequence first shows the Phase I prototype ITS (authored in the Phase I prototype ITS authoring tool). The ITS interaction is first demonstrated with one student (which is compared to another student later in the sequence). The first student is assessed by his performance in a particular scenario. The system displays the principles he could apply and those he failed to. One of the failed principles is the champagne tactic. Based on his individual, updated student model, another assessment scenario is retrieved, which he executes. He misses another enemy tactic. The system at this point looks for an ITS that addresses the whole enemy tactics group, but finds none. The student then exits and his student model is automatically uploaded to the ITS Server.

An instructor now logs on, at another machine. He is able to look at the student's performance through use of the Internet ITS Server. Now an author logs on to a third machine and what a scenario looks like to him is displayed. It primarily consists of actions and rationale (which references principles). The author also submits a Tactics ITS to the ITS server.

A new student logs on at yet another machine. He gets the same initial scenario, but a different second scenario, since he performed differently, indicating mastery of different principles, leading to a different student model, which is used as a basis for retrieval of scenarios. The second student then misses a couple of enemy tactics in a row. This time, the ITS is able to find an applicable ITS to tutor these concepts and downloads it from the ITS server and executes it, thus demonstrating our loose confederation of ITSs concept.

exercise in a scenario
some thing rights some things wrong
evaluated

I. Authoring capabilities

A. Enter each kind of knowledge required by the ITS - (scenarios, principles, topics)

B. Coordination of 2 different authors developing two related ITSs

II. AWACS ITS

Student 1

Scenario 1, Remediation A

Scenario 2 (based on kind of mistakes made in Scenario 1)

Tactics ITS

Student 2

Scenario 1, Remediation B (different mistake, different remediation)

Scenario 3 (different mistakes -> different understood principles ->
different scenario)

Send Student Models back to central site

III. Instructor at Central Site

View Student Model from central site

Add principles from central site

Go back into AWACS, see updated ITS

6.0 Future Work

6.1 Phase II

The primary goal of the Phase II effort is to develop a fully functioning authoring tool for the creation of Internet Intelligent Tutoring Systems by instructors. The Internet ITS Authoring Tool (IITSAT) will allow improved quality and efficiency of training, reduced instructor time and travel costs, and increased consistency in training while providing increased flexibility and convenience to the Air Force. Secondly, we will develop an Internet ITS for an AWACS position. In support of the primary goal there are several subsidiary ones. We will evaluate the IITSAT by having AWACS instructors use it and give formal feedback. We will develop capabilities to capitalize on the opportunities presented by the Internet. We will develop methods to allow the instructors to enter both domain and pedagogical knowledge. We will develop the required ITS reasoning capabilities. Finally, we will evaluate and improve the system. Each objective is further described below.

1. Develop a full-scale functional Internet Intelligent Tutoring System Authoring Tool (IITSAT). We will refine our methods for representing domain knowledge as an organized collection of cases and develop a friendly user interface that will allow domain experts to directly input course material and knowledge, thus authoring an ITS, without requiring significant intelligent system training. The feasibility of ITS authoring was proved by implementation of the prototype in Phase I. There are three components to this objective:

- (A) Develop Internet Capabilities. The authored ITSs will have several Internet capabilities, automatically created during the authoring process. The ITSs will send the student models to a central location for review and management by the instructor, and, keep those models up-to-date, as the student's learning progresses. The instructor will have the automatic ability to send new ITSs and updated ITS course content to the individual ITSs. Separate Internet ITSs will have the ability to coordinate their efforts in the training of a specific student. That is, one Internet ITS can request that specific training be provided to the student from another Internet ITS. In Phase I, we have already implemented prototypes of the required communication functionality and demonstrated how it can be used to support the Internet capabilities described above. This capability will be developed further in Phase II.

- (B) Develop ITS Reasoning Capabilities. We will implement the reasoning capabilities, which operate on instructor supplied knowledge, to intelligently tutor students and tailor the course of instruction to each specific student's needs and styles. The ITS will present examples, exercises, and instruction to a student, assess his knowledge and his ability to apply that knowledge in operational scenarios, build a model of the student's knowledge based on those assessments, formulate a remedial course of instruction including both additional examples and multi-media descriptions, and test the success of the remedial instruction. The feasibility was proved in Phase I by the implementation of

the prototype, which performs each of these actions and in previous ITS projects performed by SHAI. Phase II will implement the full-scale version.

(C) Develop Knowledge Entry Capabilities for Instructors. We will develop the capability for instructors to enter the knowledge into IITSAT. This knowledge includes scenarios (or cases); principles whose application is required to solve the problems in the scenarios; Topics, which are multi-media descriptions of principles and sections of the course; and Pedagogical knowledge. This knowledge includes the goals of the ITS, student action assessment methods, student mastery criteria, learning style methods, remediation method selection, and instructional strategy selection. The most difficult of these were implemented in the Phase I prototype.

2. Develop an ITS for AWACS Weapon Directors. SHAI will use IITSAT to develop an ITS for AWACS Weapon Directors, deliverable over the Internet. The ITS will instruct and assess students in the context of AWACS scenarios. The Feasibility of this objective was proven in Phase I through the development of an AWACS ITS Prototype.

3. Evaluate and Improve the system. AWACS instructors will use IITSAT and give formal feedback through a survey and questionnaire. We will also use some non-military instructors for the evaluation. Based on this evaluation, IITSAT, the AWACS ITS and its Internet capabilities will be improved. The evaluation will be repeated to assess these improvements. We will also seek to gather data on improvements in students using the resulting ITS, most likely the students will be weapons directors in training.

6.2 Phase III

There are enormous commercial opportunities for this effort, both directly and indirectly. Most directly, we will have a natural set of trainees in the form of AWACS WDs who are a direct target for this effort. The primary commercialization plan, for these ITSs, will be to demonstrate a significant increase in operational capability at the end of Phase II. The realization of this Internet ITS capability during Phase II will create a demand for similar ITSs and allow us to acquire Phase III funding to transition more Internet ITSs and the general ITS authoring tool to operational use. During Phase I, we were able to demonstrate the feasibility and usefulness of our concepts. This allowed us to line up Phase III sponsors prior to the beginning of Phase II. This is the method we have primarily used to maintain our 100% successful commercialization record. The realization of this capability during Phase II will allow us to acquire Phase III funding to transition the system to operational use.

Success in our approach will generate a broad range of potential Phase III applications. Many training domains in other military sectors as well as other government agencies and the private sector require students to develop accurate mental representations. The instructional and analytical techniques developed for this research effort and used in the ITS-authoring tool could be applied independently in regular

instruction, or in coordination with the ITS. The ITS-authoring tool itself could be enormously useful because of its ability to automatically assess the accuracy of a student's mental model of a domain. Because the ITS authoring tool is designed for quick domain application, it could be employed for new training domains with little development time. Further, we have great confidence in our approach because we have already applied the example-based ideas for ITSs successfully in other arenas.

Since IITSAT generates Internet ITSs, the rapidly expanding Internet is itself one main market segment target. Organizations who would like to generate Internet deliverable ITSs will have few choices besides IITSAT. Furthermore, the loose confederation idea promotes promulgation of IITSAT

SHAI has been involved in several Case-Based instruction ITS projects. It is now time to package our techniques and what we have learned and discovered into a general Internet ITS authoring tool. This tool would be marketed to instructors and educational software developers. SHAI already has contacts with Broderbund Software and several instructors at the high school, college and private industry level. From our previous ITS projects, we also have contacts with government departments responsible for developing training systems.

The ultimate product of this research is a tool which can be used by domain experts to create example-based instruction ITSs. There are two types of products for commercialization. First, we can sell the Internet ITS authoring tool itself, to allow the purchasers to develop their own ITSs. SHAI has a lot of experience in this area. Many of our software products are made to be customized by domain experts with no programming experience. These tools include AMP, a planning and scheduling tool, EnvKB, an environmental design tool, and AFTES, a document generation tool. The tools, while allowing non-programmers to create intelligent applications, also have the flexibility and access to allow serious developers to interface the resulting applications to other software modules and to significantly modify the application itself. Similar ease of use for non-programmers and flexibility for software developers will be included in the resulting ITS tool and architecture. From our product commercialization experience, we understand how to successfully market such a product. Articles and advertisements in relevant journals and magazines, exhibition booths and tutorials at relevant conferences, direct mailings using lists purchased from relevant organizations, involvement in industry associations, and a strong World Wide Web page are all important aspects. These methods generated an order of magnitude more funds than SHAI received in SBIR funds for some products.

Esteem Software Incorporated (ESI), a leading supplier of Case-Based Reasoning Development tools, plans to market IITSAT. We have a long history with ESI, having been contracted to develop their flagship product, ESTEEM, in 1991. ESTEEM has been remarkably successful, as described below. We expect that ESI, using their considerable market resources, can successfully market IITSAT as well.

Secondly, we can make use of the tool to create ITSs for individual clients and markets. Because the ITS is designed for quick domain application, it could be employed for new training domains with little development time. Marketing ITS creation services is similar to SHAI's core business of marketing AI research and development services, at which we are very successful. SHAI's current revenue from services is approximately \$2.5 million and will be close to \$4 million in 1998.

For example, we have been well-received by Sterling Wiggins and Lloyd Pollock of Oracle's Training Division, about the use of the authoring tool to develop of ITSs to teach Oracle applications, especially database administrator concepts. Additionally, since many organizations write applications on top of Oracles products, they are interested in selling a customized version of the authoring tool so that those application developers can also develop ITSs for the applications that they have developed.

Marketing ITSs to individual, vertical marketing segments will be highly dependent on the markets involved. For example, SHAI has developed relationships with several educational software companies and text book publishers. This is in preparation for ITS aimed at high school and remedial college math (algebra, geometry, trigonometry, and calculus) and science (physics, chemistry, and astrophysics). Additionally, SHAI has good contact with customers in the military likely to purchase developed ITSs, including the Dismounted Infantry School at Fort Benning, the Armored Infantry School at Fort Knox, the AEGIS Program Office, the AEGIS Training Center, AEGIS Training Group, Surface Warfare Officer School, Army STRICOM, and SPAWAR.

Significant improvements in the performance of AWACS WDs will be used as a basis to promote the system to other military domains.

References

- Aleven, V. and Ashley, K.D. (1992). Automated generation of examples for a tutorial in case-based argumentation, Intelligent Tutoring Systems '92 Proceedings, Montreal, Canada
- Bloom, B., S., The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher, 13(6), 4-16.
- Gragg, C. I., "Because wisdom can't be told", Harvard Alumni Bulletin, October 19, 78-84, 1940
- Edelson, D.C. (1992) When should a cheetah remind you of a bat? Reminding in case-based teaching. Proceedings of AAAI-92, San Jose, CA.
- Kolodner, J., and Simpson, R. (1986). Using experience as a guide for problem solving. Machine learning, a guide to current research. Boston, MA Kluwer Academic Publications.
- Regian J. W., & Shute, V.J., "Evaluating Intelligent Tutoring Systems", In E. L. Baker and H. F. O'Neil, Technology assessment in education and training, Hillsdale, NJ: Lawrence Erlbaum, 1994
- Rosenblum, J.A. and Moore, J.D. (1993). Participating in instructional dialogues: finding and exploiting relevant prior explanations. Proceedings of AI-ED, Assoc. Adv. Comput. Educ., Charlottesville, VA.
- Schank, R., and Abelson, R. (1977). Scripts, plans, goals and understanding inquiry into human knowledge structures. NY: Erlbaum.
- Schult, T.J. (1993). Tutorial reminders in a physics simulation environment. Proceedings of AI-ED, Assoc. Adv. Comput. Educ., Charlottesville, VA.
- Shute, V.J., "Aptitude Treatment Interactions and Cognitive Skill Diagnosis", In Regian J. W., & Shute, V.J., Cognitive Approaches to Automated Instruction, NJ: Lawrence Erlbaum, 1992.
- Shute, V.J, Glaser, R., Raghavan, K., "Inference and discovery in an exploratory laboratory", In P. L. Ackerman, R. J. Sternberg, R Glaser, Learning and individual differences, San Francisco: Freeman, 1989.
- Stottler, R. H., Henke, A. L., & King, J. A. (1989). Rapid retrieval algorithms for case-based reasoning. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Detroit, Michigan.

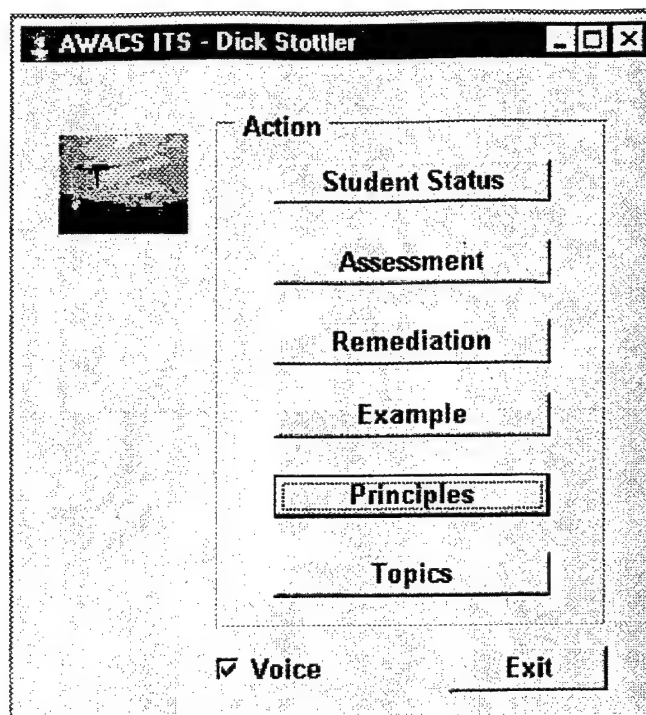
Towne, D.M., Munroe, A., Johnson, M.C., "Generalized maintenance trainer simulator: Test and Evaluation" (report No. NPRDC-TR-83-28), Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, 1983.

Towne, D.M., Munroe, A., "RAPIDS: a simulation-based instructional authoring system for technical training" (report No. 112), Los Angeles, CA: Behavioral Technology Laboratories, University of Southern California, 1989.

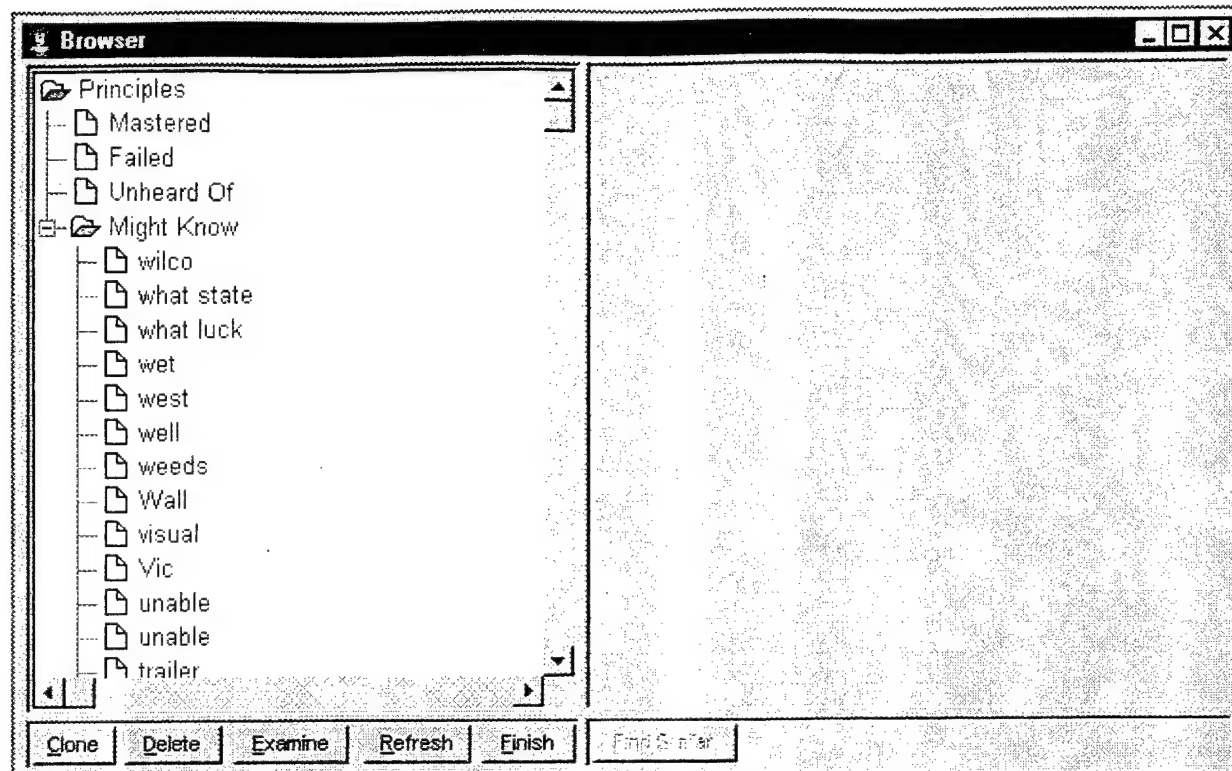
Whitehead, A. N. The aims of education. New York: Macmillan, 1929.

Appendix: Demonstration Screen Dumps

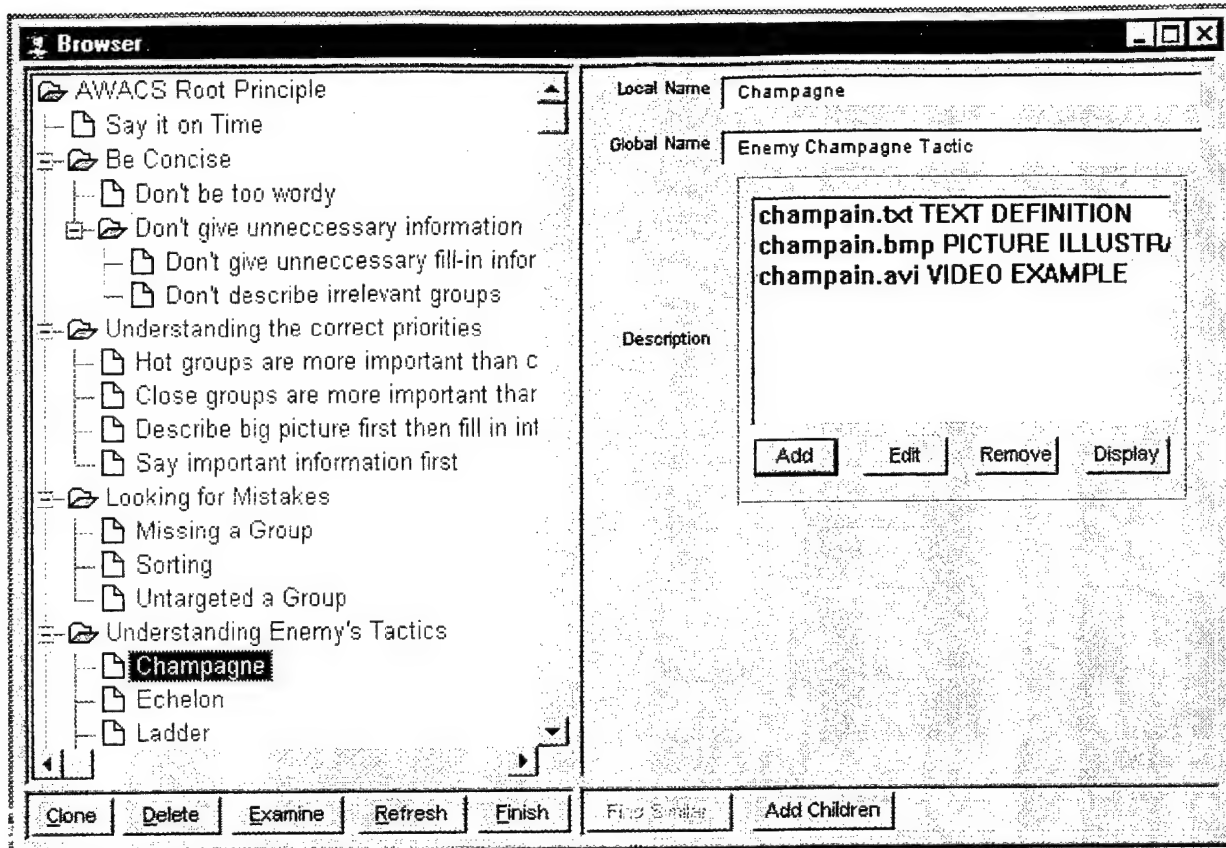
The first screen dump, below, shows the main student screen which appears after student, "Dick Stottler", has logged on to a computer with a valid Internet connection.



The "Student Status" button shows which Principles have been mastered, failed, unheard of, or might be known. At this point, since Dick Stottler is a new student, all principles have defaulted to might know, as shown below.



The student can also select the "Principles" button on the main screen to see the hierarchy of principles taught by the ITS. By selecting a principle, such as champagne, the available description files are shown on the left, as shown below.



The corresponding text and bitmap descriptions are given below.

champain.txt:

CHAMPAGNE - Attack of three with two in front and one behind. Lead two groups are attempting to bracket with the trailing group flying up the middle.

champain.bmp:



The "Assessment" button on the main screen, activates an AWACS scenario simulation and assesses the students actions in it. If the Voice option is checked (which is typical), the students actions consist entirely of voice inputs through an attached microphone. If it is not checked, inputs are typed in.

After assessment is selected, since Dick Stottler is a new student and all principles need to be tested, the system automatically and dynamically selects the scenario which tests the most principles. The brief for this scenario is displayed and reproduced, below.

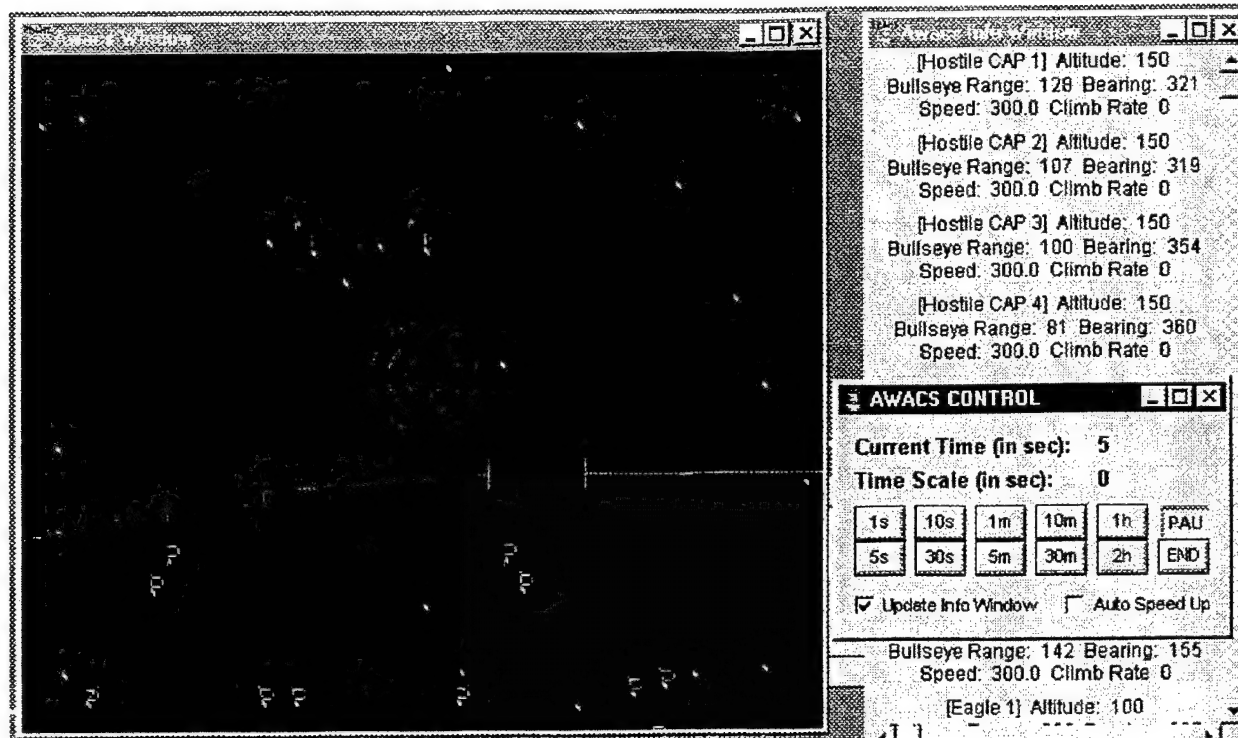
Enemy Territory is to the north, as shown by the border.
The attack will occur in two simultaneous strikes.

The East Strike consists of Eagles 1-3 and Wolves 1-3.
They will marshall then proceed North to their indicated targets.

The West Strike consists of Cougars 1-3 and Rambos 1-3.
They will marshall and proceed North to their indicated targets.
Both groups will egress south through the indicated airways.

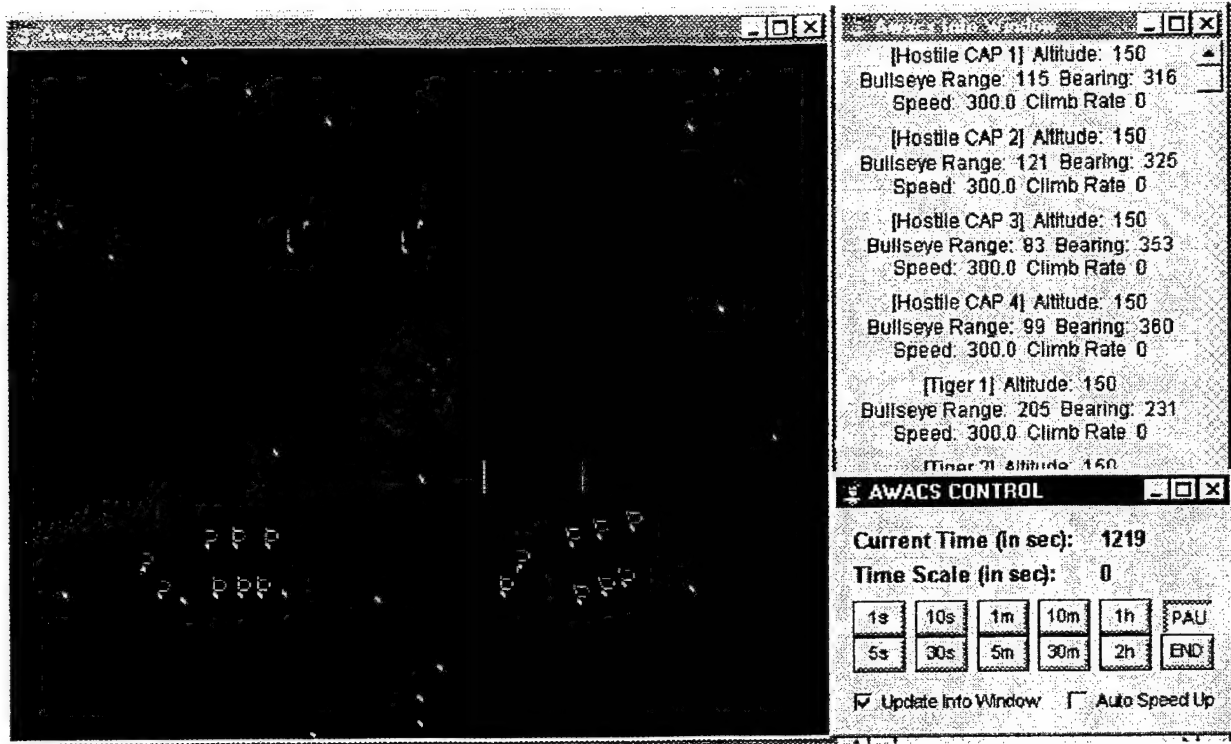
Two friendly CAPs, Tiger 1 & 2, will defend the east
at (90,320) and (80,320), respectively.
Two CAPs, Tiger 3 & 4, will defend the West at (300,320)
and (310, 320) Respectively.

After the brief, the scenario starts. The simulation is made up of three windows as shown below. The primarily black window is an approximate recreation of the AWACS radar display with which the weapons directors are familiar. The yellow dots represent radar returns, including noise, which is often called "sparkle".

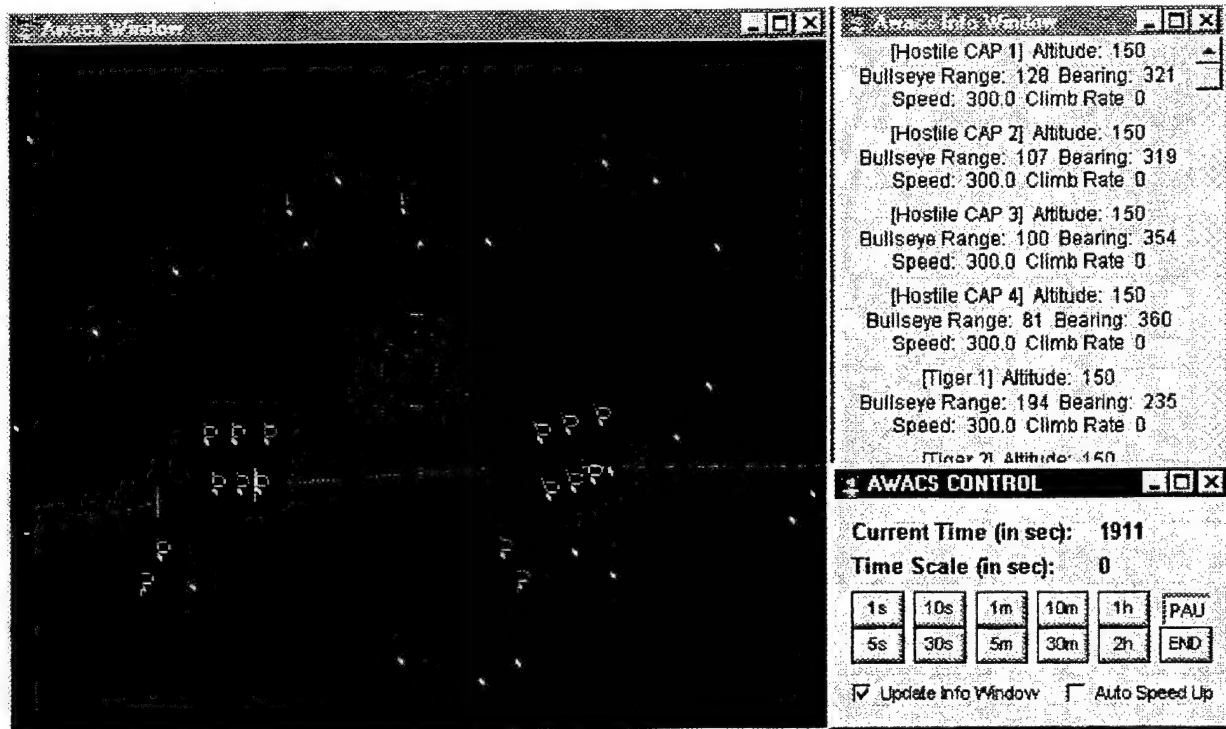


In this scenario, the red Xs in the North denote strike targets. The purple concentric circles represent the bullseye coordinate, from which most communications are referenced. The blue star represents the current own-plane position. The white line represents the border between opposing forces with the gaps representing predesignated egress routes. The other icons are standard Air Force symbols. The line protruding from each is its velocity vector. At this time there are four enemy CAP in the North. There are four friendly CAP just south of the border and two separate strike packages are marshaling in the southernmost part of the screen, each consisting of an OCA screen followed by strike aircraft as can be seen more clearly, next page.

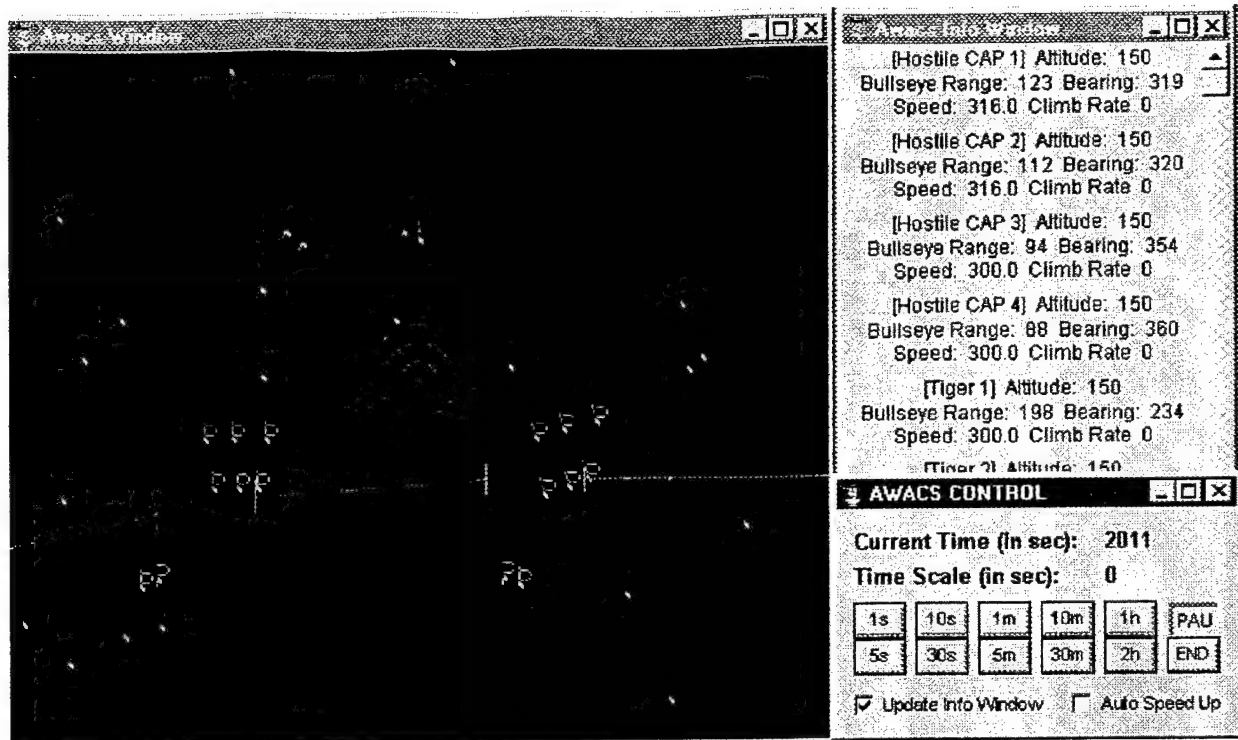
There are two other windows. Shown in the foreground is the Simulation control window. Its main function is to allow the student to change the rate of progression of time, primarily for demonstration purposes. The last window is the information display window. It shows bullseye range and bearing, altitude, and speed for every group in the scenario. This window can be brought to the foreground and/or scrolled to see the information for each group.



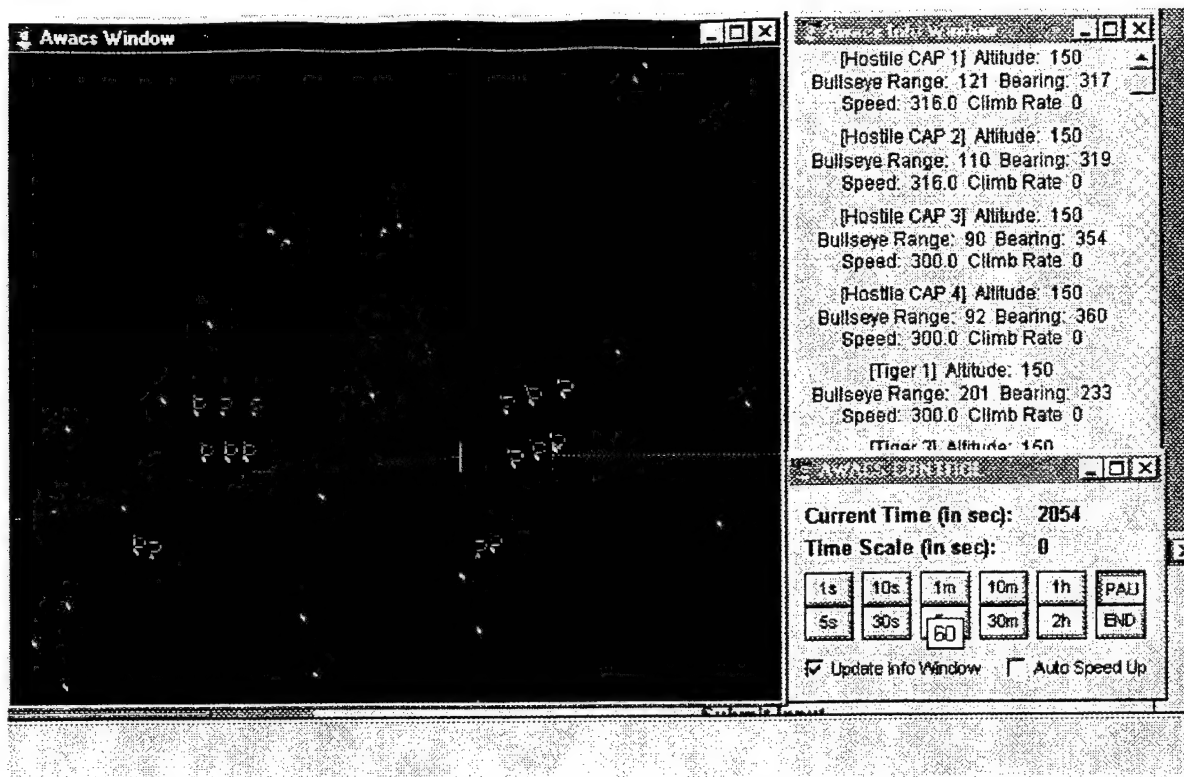
At about 1900 seconds into the scenario, Eagle One requests a picture with the call "Eagle One, Picture" which is broadcast to the student from speakers. The picture below, shows the scenario at this time. The student should respond with, "Bandsaw, Eagle one, two counter-opposing CAPs azimuth split 10, east CAP bullseye 319, 107, estimate 150.



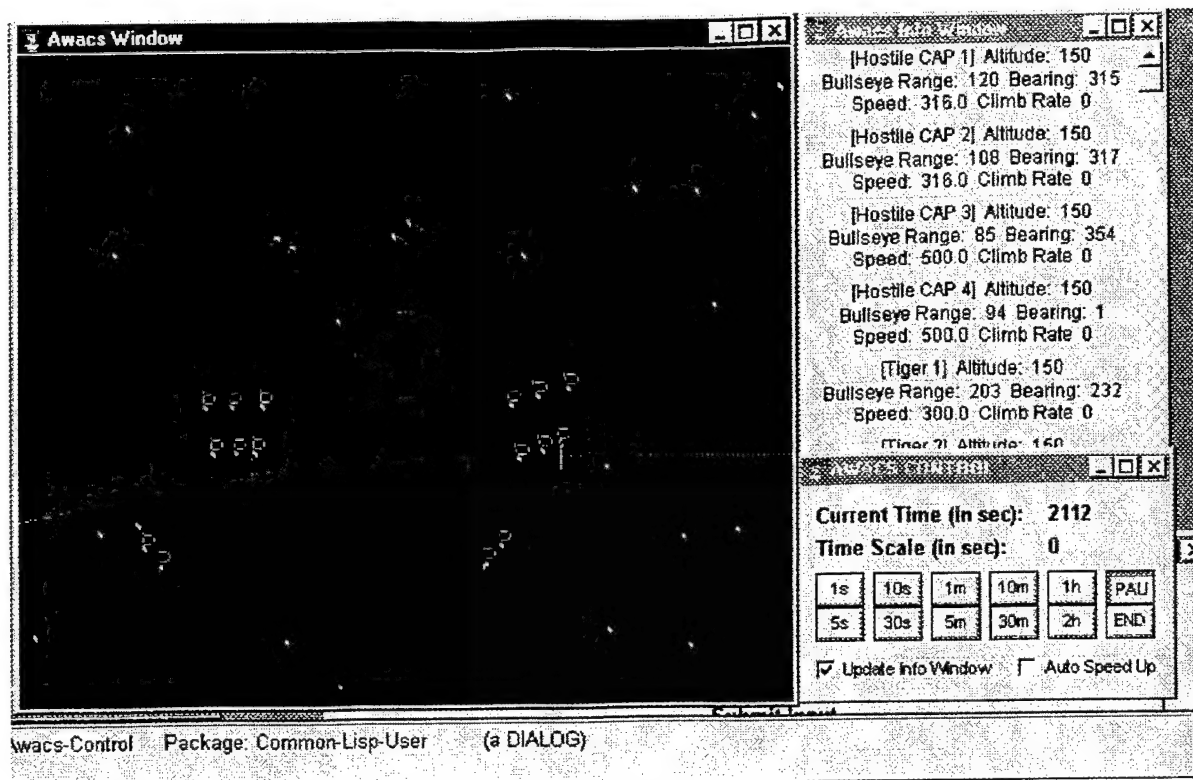
At about time, 2000 seconds, the left hostile CAPs commit and the student should say, into the microphone, "Bandsaw, Eagle One, both CAPs committed." The screen dump, below, shows the scenario, at this time.



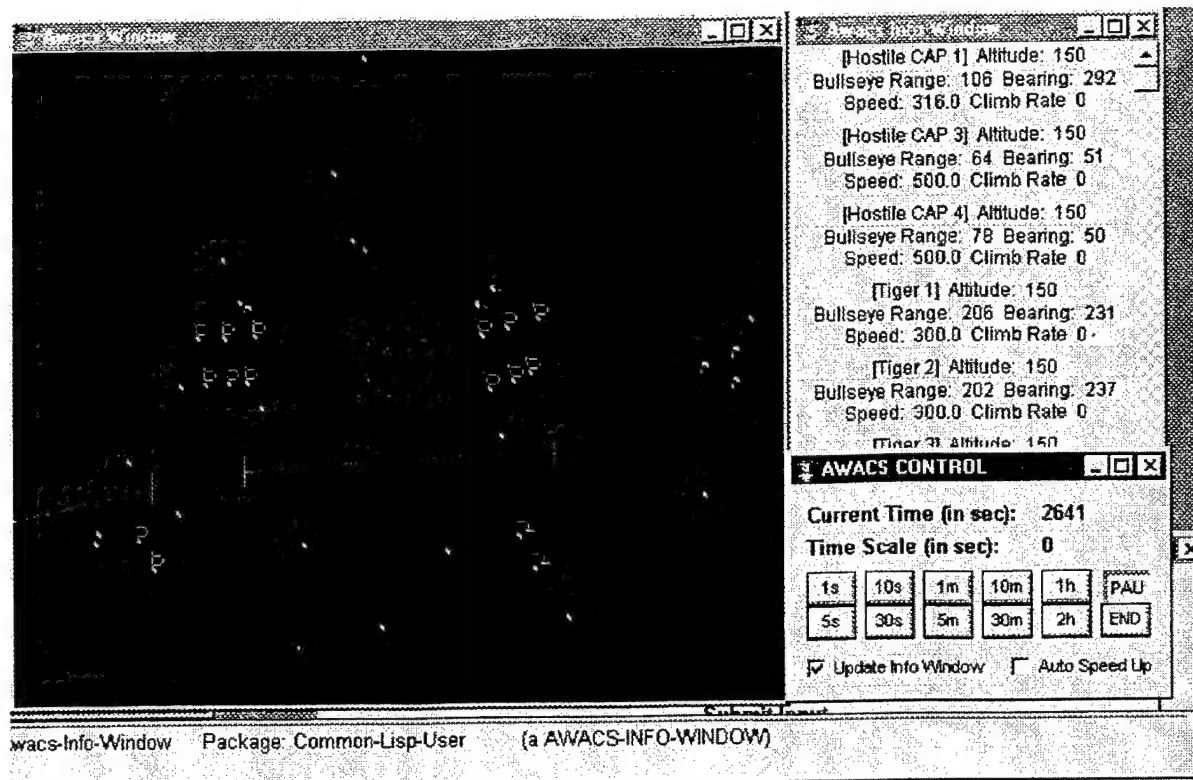
Later at about time 2050, the student receives the call, "Cougar one, picture" with the situation shown below.



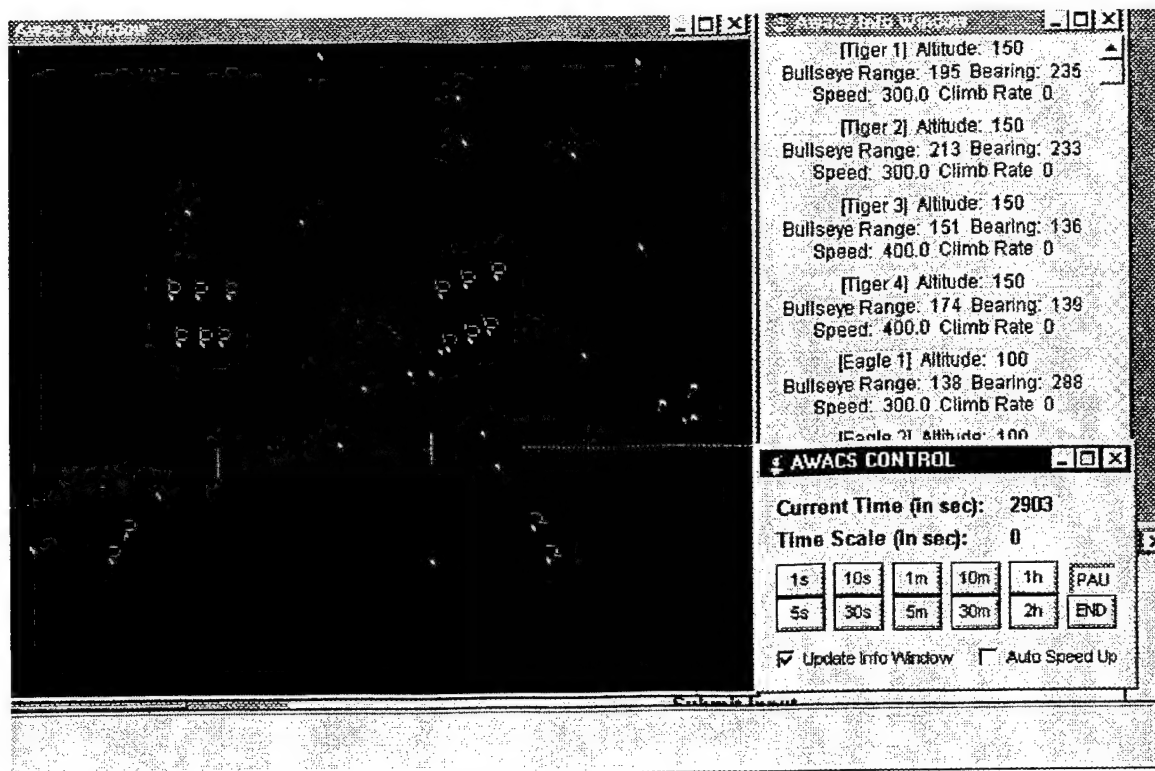
The student responds with, "Bandsaw, Cougar One, two counter-opposing CAPs, azimuth split 10, east CAP bullseye 360, 81, estimate 150." At about 2100 seconds the right enemy CAPs commit, as shown below, and the student should state, "Bandsaw, Cougar One, both CAPs committed"



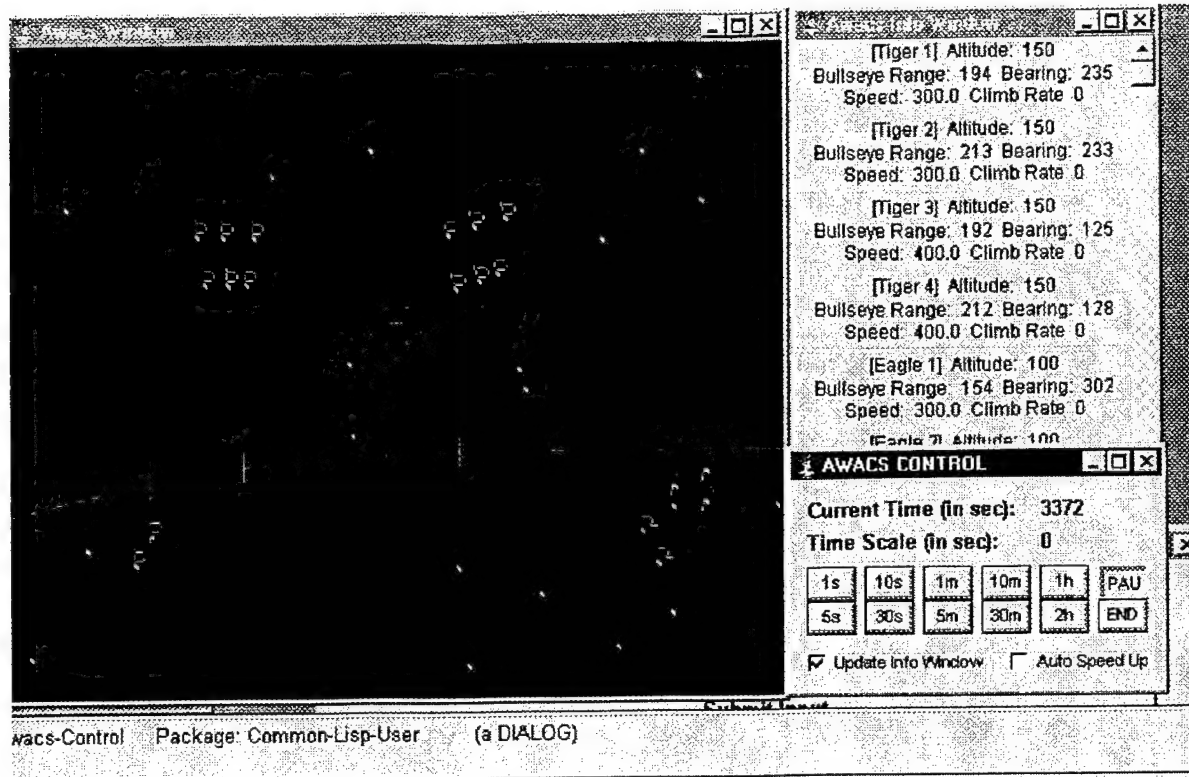
The situation at about 2640 is shown below. The enemy CAP have intercepted the friendly screens both left and right and are engaged. One of the Enemy CAP has been destroyed and the other three are about to be. Meanwhile a new formation of three groups has appeared.



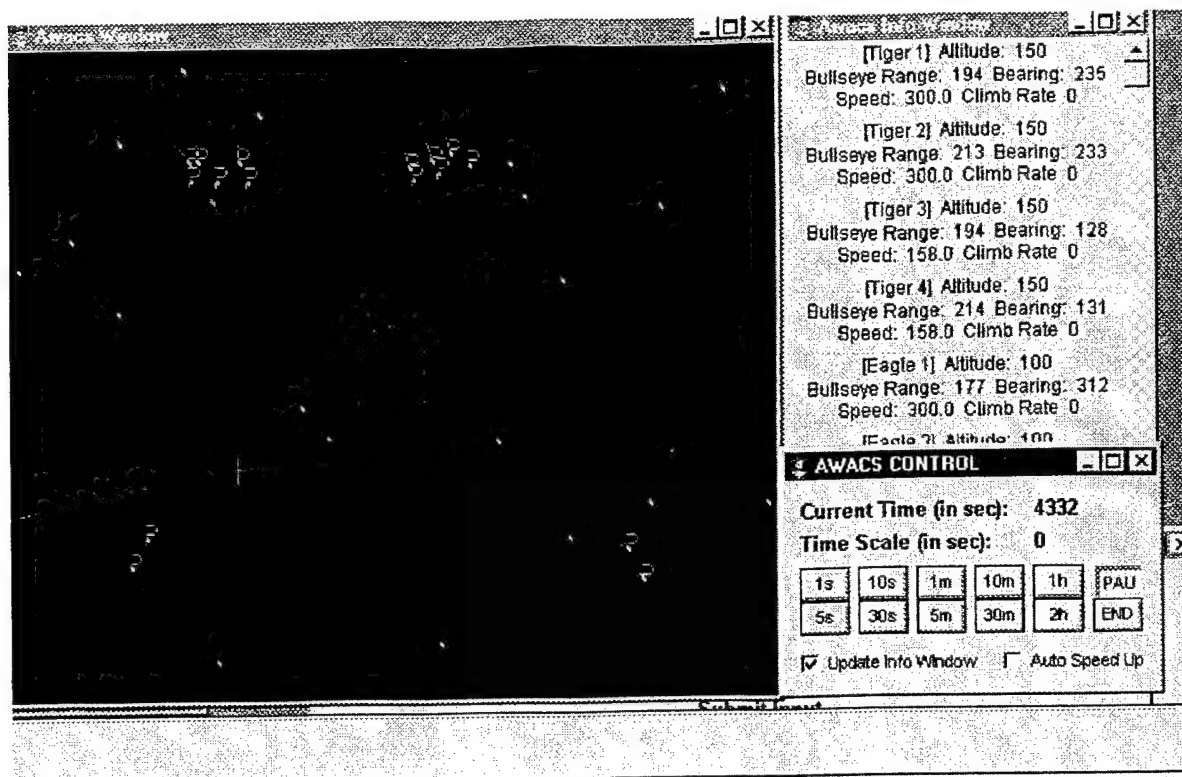
Later, at 2900, the student receives another picture call. He hears, "Tiger One, picture" and sees the following:



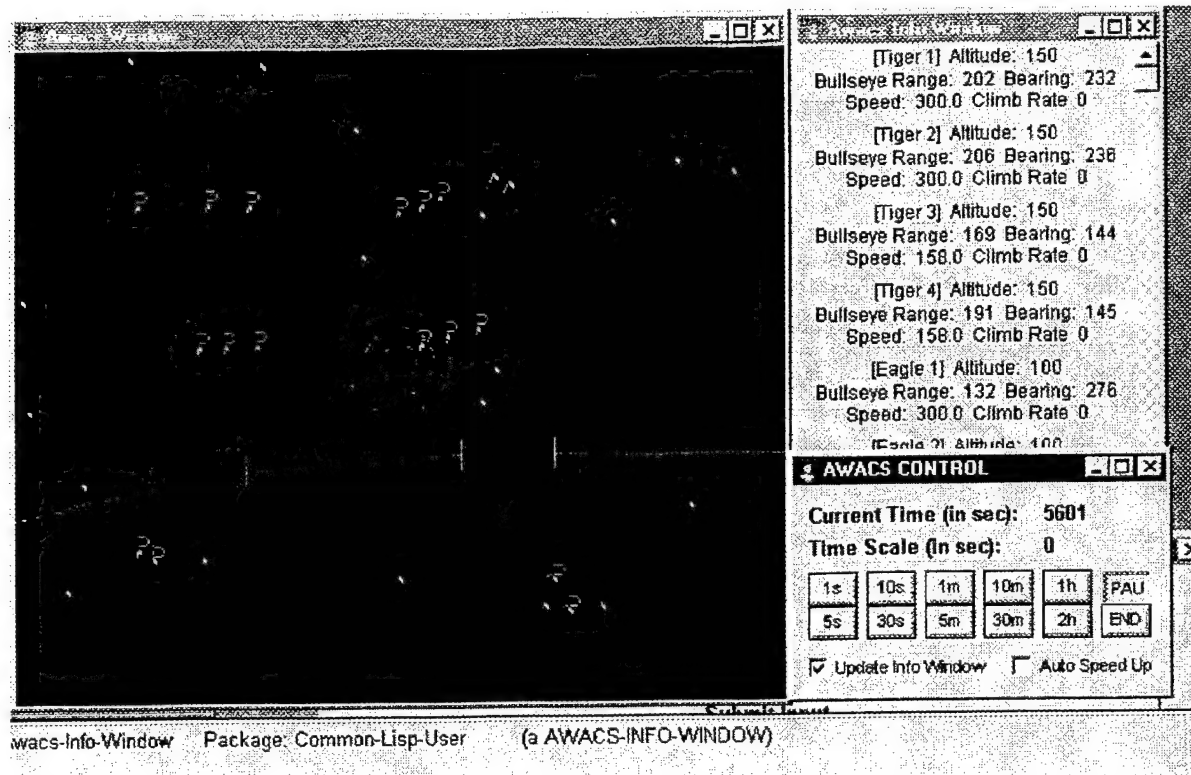
He should respond with "Bandsaw, three group champagne, azimuth split, twenty, west lead group bullseye, one zero three, two hundred eleven". But for demonstration purposes he'll say "Vic" instead of "champagne". Later, at around 3600, those groups are destroyed. The screen dump below, shows the situation just before their destruction.



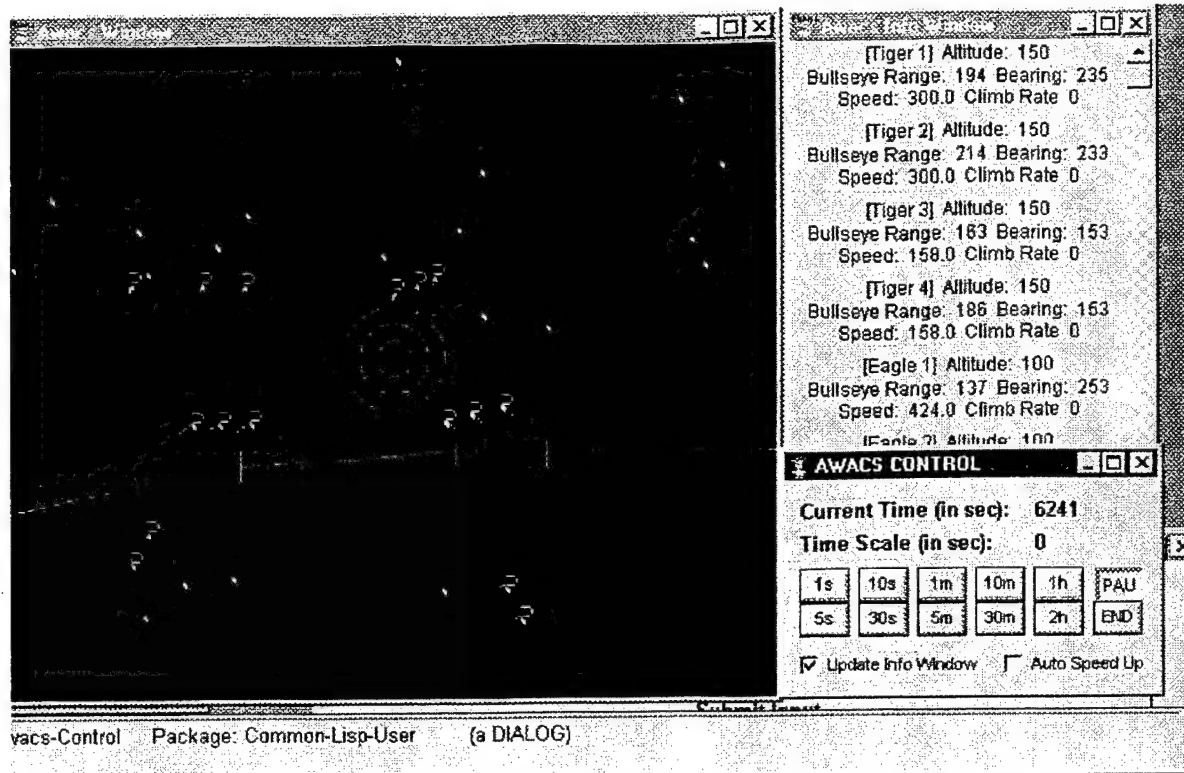
At around 4100 seconds, the screening aircraft turn back and cross the strikers, who are still outbound, as shown below at 4330.



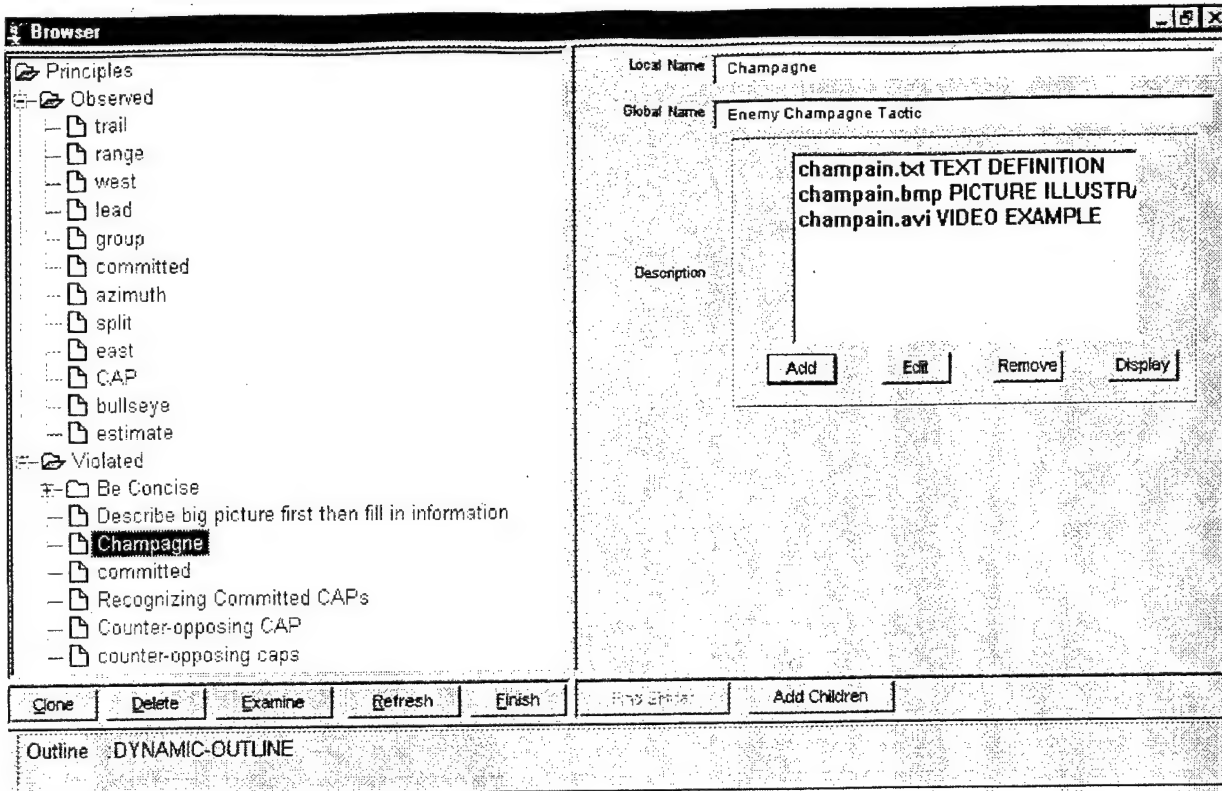
At around 5300, two groups appear and give chase to the strikers, at around 5600, they are close enough to become tactically significant as shown below, and the student should make the call, "Bandsaw, Rambo 3, 2 groups, group bullseye 27, 129, group bullseye 32, one thirty."



Those groups are destroyed and the screening and striking aircraft egress through the indicated airways.

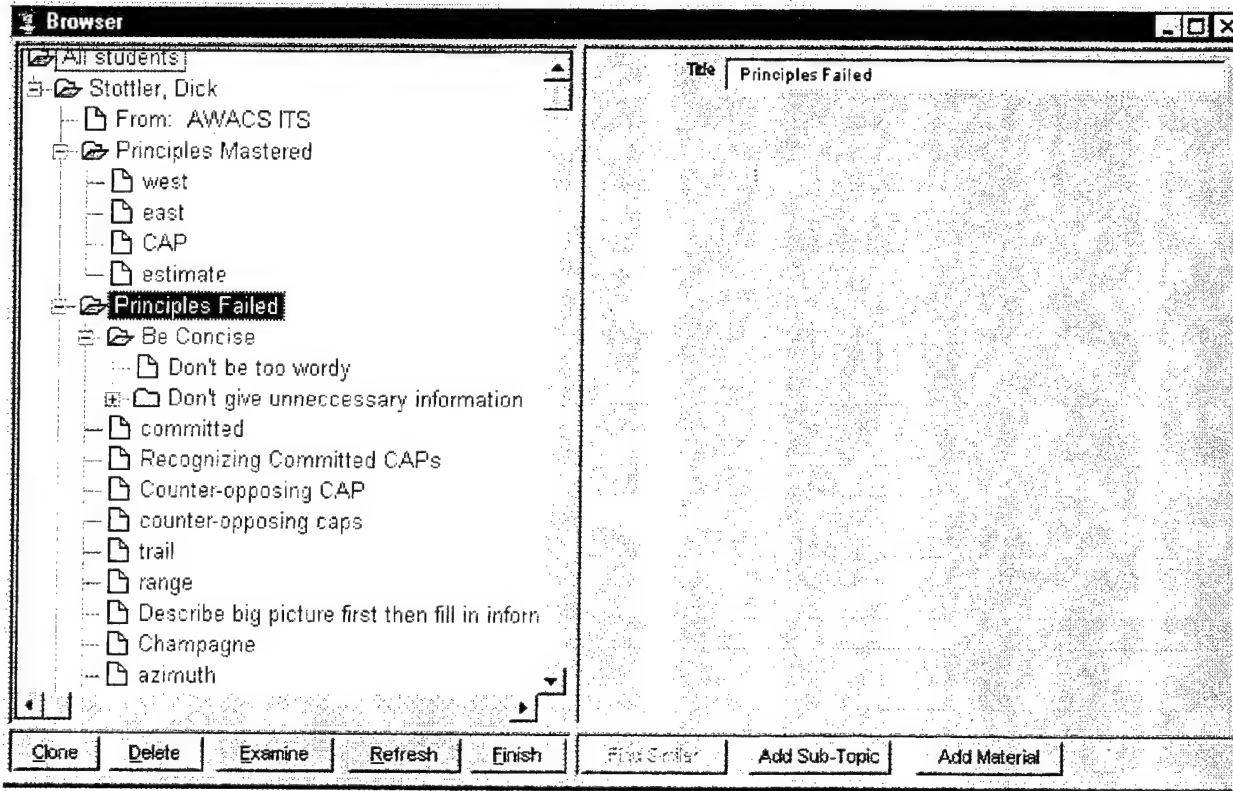


When the scenario ends, the principles that the system infers the student understands and can apply, are displayed to the student, along with the principles violated by the student. These inferences are based on observation of actions requiring application of those principles. Further, the student can highlight a principle and view various descriptions of it as shown below.

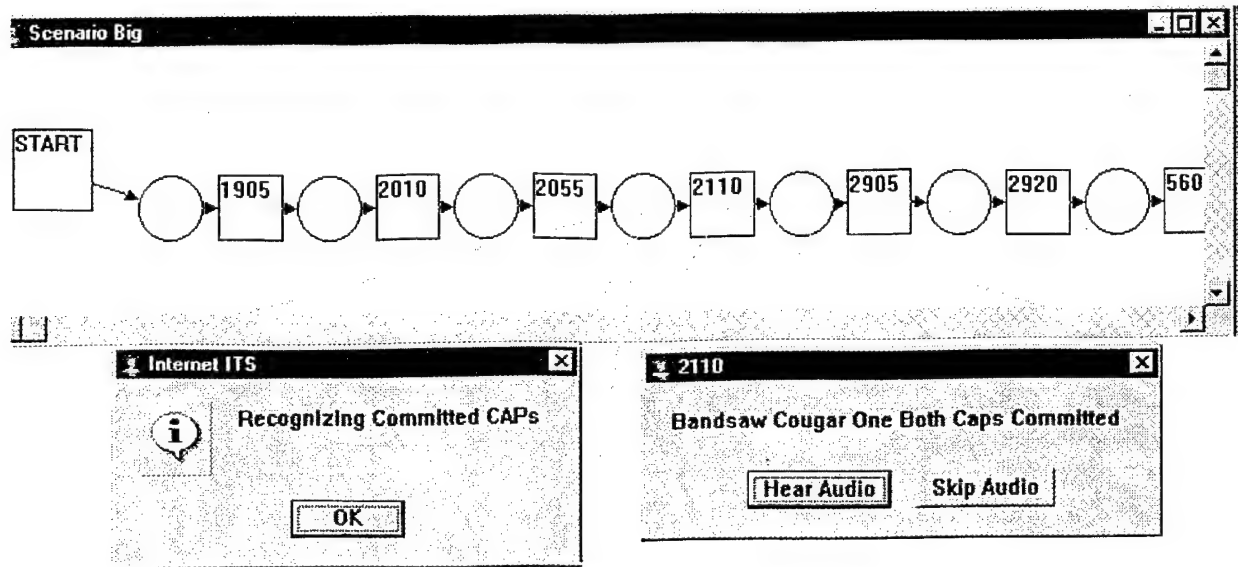


The student then goes back to the main screen and presses the "Assessment" button again. The system decides which scenario would best fit his current assessment needs and happens to choose scenario 2, because it seeks to test the student on knowledge of other enemy tactical formations, since he got one wrong and many others are untested. A similar sequence transpires as described above. For demonstration purposes, the student misses the description of the tactic. After missing two separate enemy tactics, the system believes that he has a general deficiency in his knowledge of tactics, unfortunately at this point there are no other ITSs to handle this deficiency. The student then exits and his student model is submitted to the server automatically.

The demonstration sequence proceeds by logging on as an instructor from another computer, connected to the Internet. One common operation that an instructor might want to perform is to check a student's progress. By clicking on the "Retrieve Student Model" button, the instructor can get Dick Stottler's student Model, including a description of what principles he understands and can apply, those that he failed to apply and those which have not yet been tested.



The instructor might want to author a new ITS or update an existing one. To do this requires inputs relating to three items - scenarios, principles, and topics. Principles and topics can be entered organized, and updated in reference to the hierarchies shown throughout this demonstration sequence. Entering scenarios is shown by selecting the "Edit Scenario" button, which brings up scenarios in the form shown below. By clicking on squares (actions) the ITS author can see the expected text and play an audio version. By clicking on a circle (rationale) the author can see the principles whose understanding is required and demonstrated by the action.

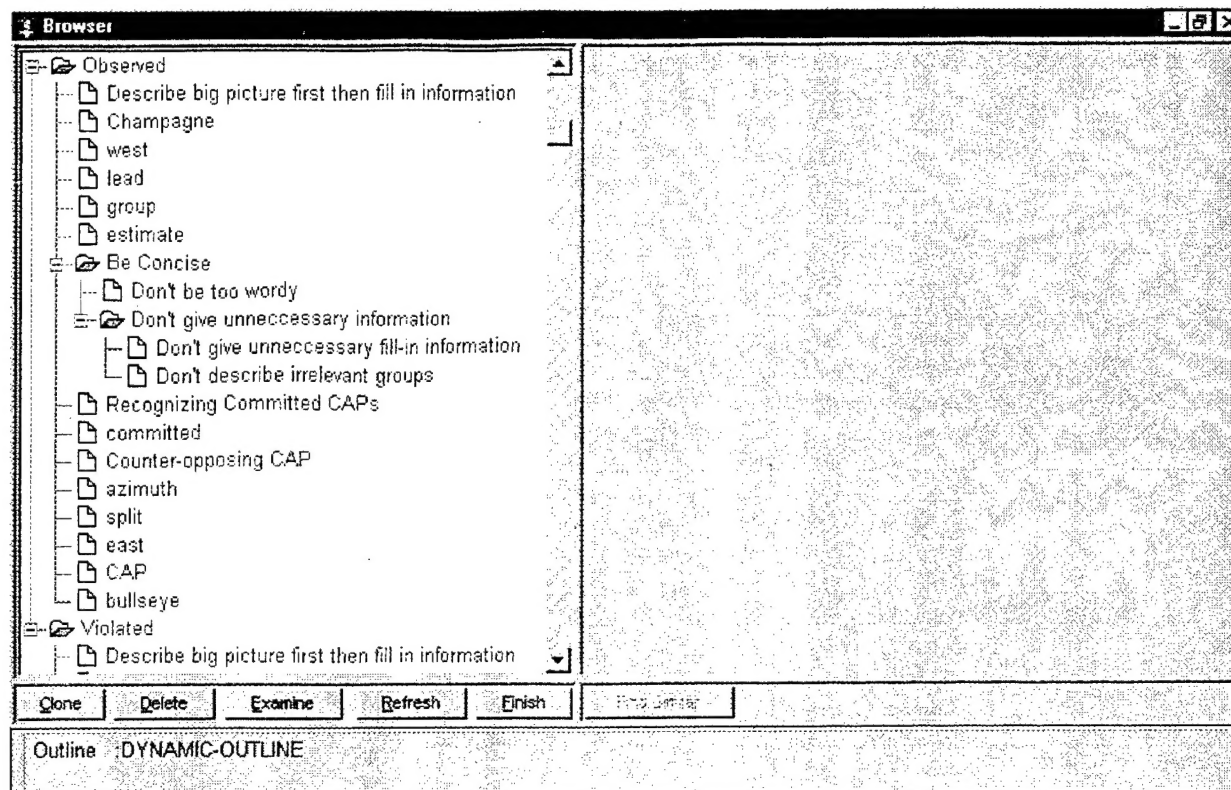


After an author has created an ITS, he would want to submit it to the ITS server so that it can become automatically available, when needed. In this demonstration, we'll add a very simple ITS to tutor tactics specifically. It will be called automatically whenever an ITS determines that a student is generally weak in his understanding of tactics. To submit an ITS, the author selects the "Submit an ITS" and enters the following information in the dialog boxes.

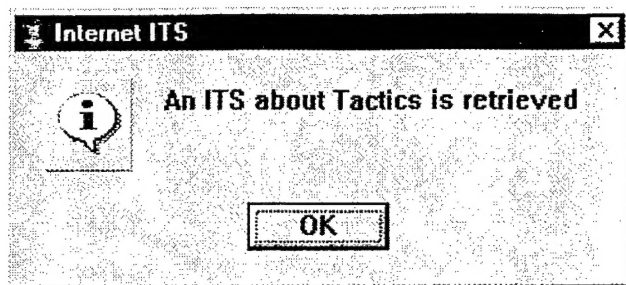
Name: Tactics ITS
Start Command: tactic-its
file: tact-its.bil

By selecting the "Retrieve ITS" button from the instructor/author's interface, a list of available ITS is retrieved from the server. That list now includes the Tactics ITS, which it had not previously.

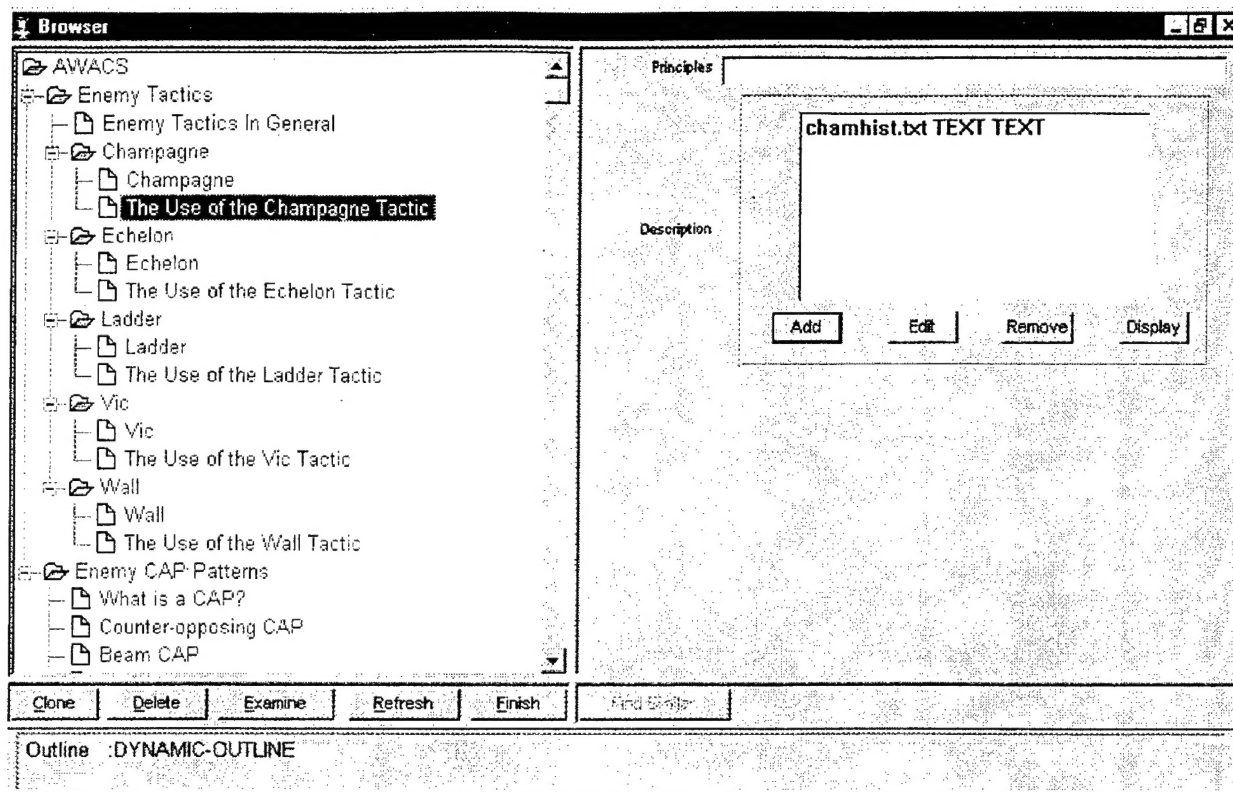
The demonstration sequence continues by logging in as another student - Ricky Tang. He selects the "Assessment" button. Since this is also a new student, needing testing in all principles, he gets the same initial scenario as the first student. This second student does much better than the first student as this debriefing, below, shows. He has performed differently than the first student. In particular this second student has gotten the champagne tactic principle correct.



Because the second student performed differently, he has a different student model and receives a different second scenario. This student will be assessed next on Scenario 3 (while the first student received scenario 2, at this point). For demonstration purposes this second student misses the enemy tactic in the scenario, get another scenario for assessment, and misses another enemy tactic. This prompts the ITS to assume that the student has a general problem with enemy tactics and it attempts to find an ITS to address this problem. This time it finds one at the ITS server, that an author recently added. For the first student, this ITS did not yet exist. Note that we did not explicitly inform the second student's ITS that a new ITS existed or even have to be aware that a previous ITS, which might want to use ours, existed. It automatically, dynamically looked for and found it. This demonstrates the loose confederation of ITSs concept. This fact is displayed for the student and shown below.




The student can also access an "electronic text book" by selecting the "Topics" button. The Topics are arranged hierarchically, in a form similar to traditional text books. Each Topic describes some logical portion of the course and can be broken into smaller topics. Presumably most of the topics described principles that the student needs to learn and apply. The partial topics hierarchy is shown below.



The main student screen also includes a "Remediation" button, which gives him the list of failed principles. There is also an "Example" button that allows the student to select principles which are used to retrieve a scenario that illustrates those principles. That scenario is then played to the student by the ITS. The ITS plays the role of WD. I.e. it speaks the WD's lines. This sequence looks very similar to the scenario screen dumps given at the beginning of this sequence.

Certification of Technical Data Conformity (May 1987)

The Contractor, Stottler Henke Associates, Inc., hereby certifies that, to the best of its knowledge and belief, the technical data delivered herewith under Contract No. F41624-97-C-5017 is complete, accurate, and complies with all requirements of the contract.



Signature

Christopher J. Rognier

Name - printed

V. P. of Administration

Title

2/6/98

Date